Learning Low-code: Potential of Mendix Platform in Building Applications

Giovani Francisco

Haaga-Helia University of Applied Sciences
Haaga-Helia Bachelor’s Degree
2024

Abstract

Author(s)
Giovani Francisco

Degree
Bachelor of Business Information Technology

Report/Thesis Title
Learning Low-code: Potential of Mendix Platform in Building Applications

Number of pages and appendix pages
75 + 1

This thesis presents a comprehensive analysis of learning Mendix, a low-code platform. It documents the learning journey of the trainee working with the platform over the course of several weeks. The observation spans an 8-week monitoring period from October 21 to December 13, 2024.

The primary objective is to develop a comprehensive understanding of Mendix and its development practices, including the areas of domain modelling, microflows, and security configurations. A professional analysis of work experience, documenting challenges faced, strategies, and accomplishments, is monitored to provide wisdom into the trainee's growth throughout the process.

The study focused on key concepts such as Scrum methodology, low-code development, Mendix, and utilizing the Mendix Studio Pro platform in building web applications. In addition, the study highlighted the critical importance of data validations in ensuring integrity and security of the application through validation checks. This also emphasized on the significance of user experience design, demonstrating how interfaces and user-centric designs contribute to building applications that are not only functional but also user-friendly.

The thesis includes daily monitoring of work-related activities and weekly analysis, identifying the technical strengths and areas for improvements. Throughout the project, the trainee reflected on both successes and difficulties encountered while working with the Mendix platform, allowing for continuous self-assessment and adjustment. The thesis helped identify patterns in the trainee's progress, leading to a better understanding of how to approach complex development tasks and meet project objectives efficiently.

The thesis provides insights and guidance for those who are interested to learn low-code platforms, specifically Mendix. The thesis serves as a valuable tool for learning, allowing for continuous improvements and ultimately resulting in a more competent and confident developer. This offers practical guidelines on how to navigate common obstacles in low-code development while highlighting the importance of key concepts in Mendix. The thesis offers a firsthand perspective on how to apply theoretical knowledge to practical projects, helping build a foundation for success in the field of low-code development.

Keywords
Low-code, Mendix, Domain Modelling, Microflow, Scrum Methodology, Mendix Studio Pro, Data Validation, User Experience

Contents

1 Introduction

This is a diary-based thesis of a junior software developer, referred to as the ‘trainee’, that documents the experiences and skills acquired over a specified timeline. This diary thesis covers an 8-week observation period from October 21, 2024 - December 13, 2024. The diary details how the trainee pursues self-discipline and achieves professional growth through work experience.

1.1 Company

Tietoevry is an Information Technology (IT) software and service company that provides different kinds of services such as engineering services and business consulting and outsourcing. The company has over 24,000 experts across 20 countries that specializes in different products which provides services to its customers in approximately 90 countries. It was established in 1968 and Espoo, Finland is its main headquarter (Tietoevry 2024a). The company’s annual revenue for 2023 was about 2.85 billion euros, and the company target revenue is to rise for 0-3% further at the end of year 2024 (Tietoevry Corporation Interim Report 2024, 3).

The operative management of Tietoevry consists of the President and CEO who is appointed by the Board and is also responsible for the internal efficiency and quality of the company. The president and CEO is assisted by the Executive Management which consists of Head of HR, Head of strategy and CFO, and the different heads of the company’s five specialized line of business (Tietoevry 2024a).

Tietoevry has made big changes throughout the years internally especially in its line of business of six down to five. These lines of business are designed to address different customer-related problems which aims to create business transformation that have full operational responsibilities of its own while scaling up the company. Tietoevry’s five lines of businesses are Tietoevry Care, Tietoevry Create, Tietoevry Industry, Tietoevry Tech Services, and Tietoevry Banking.

1. Tietoevry Care
   Tietoevry Care is transforming the Nordic health and social care sector through modular, open, and interoperable software, placing citizens and patients at its core. Its solutions strengthen collaboration between health professionals, colleagues, and patients, while enhancing user experience, job satisfaction, and the quality of care (Tietoevry 2024b).

2. Tietoevry Create

The business is the premier digital accelerator focused on innovation and sustainability. The company delivers agility and speed in creating data-driven, cloud-based software solutions that provide competitive edge (Tietoevry 2024c).

3. Tietoevry Industry
   Tietoevry Industry specializes in industry-specific software and data integration services. The business software is meant for close collaboration across a wide variety of customers such as public sector, pulp, paper and fibre, and energy and utilities segments (Tietoevry 2024d).

4. Tietoevry Tech Services
   The business promotes modernization across processes, applications, and infrastructure by leveraging automation, cloud and data technology. It offers comprehensive IT solutions and services ensuring business success (Tietoevry 2024e).

5. Tietoevry Banking
   Tietoevry Banking offers its expertise in the financial institutions and banks through accelerated services. Through accelerating digital banking, the software aims to reduce complexity and focuses on delivering top-notch quality service (Tietoevry 2024f).

1.2 Job Profile and Responsibilities

Working under the company's Tietoevry Create line of business, the trainee's role description is a Software Developer. Engaging with projects related to web development whether internally or customer-related projects. Moreover, the trainee collaborates with the assigned team to deliver solutions based on carefully analyzed customer problems and requests.

The trainee is required to have a background on the programming language the team is using, usually Java and JavaScript. Knowledge about REST API, Git and GitHub is also highly essential. Participation in daily, weekly and monthly meeting are also advised for the trainee, especially if it is highly related to the project at hand. Information about Scrum, UI Design, Agile Development and User Experience Design are also suggested. Resources are provided by Tietoevry to further elevate this knowledge and build competences of its employees.

Involvement with the project concerning Low-code, specifically Mendix, the trainee at least must have the desire to learn it apart from the traditional way of building web applications. Learning

Academy offered by Mendix boast a ton of new knowledge that are divided into series of paths which then enables the trainee to get certificates of completion at the end of every series.

During the start of the project, the trainee has been involved in multiple Proof of Concepts (PoC) presented to the target customer. Participated in daily meetings online and on-site, acted on the requests and presented the updates and changes to the customer.

1.3 Required Skills

Going through the learning paths that are offered by Mendix is mandatory for the trainee to know how the software works. During PoC phase, obtaining the necessary knowledge and minimum requirements from Mendix were done at the same time.

The required skills to make use of the software from Mendix is to obtain certificates as proof of competence. Certificates offered by Mendix consists of Rapid, Intermediate, Advanced and Expert. A prerequisite of finishing the Beginner path is needed to proceed with the certifications. The trainee has already obtained Rapid and Intermediate Certificates. Though one can still obtain the Intermediate Certificate without finishing the entire course, the trainee aims to finish the courses covered in Intermediate Certification before proceeding. Background in certain languages such as Java and JavaScript are needed when working with a more complex projects and requirements that requires the use of multi-platforms.

A crucial part of the trainee's role is to know about Agile Development. Mendix platform is driven totally using an Agile approach. In every project, information about Usability, UI Design and Accessibility is valuable when implementing designed user interfaces. Working knowledge of Scrum and use of the built-in framework of Mendix in implementing these Agile methodologies are beneficial which are also part in the learning academy.

On top of these, communication is needed for the development of these technical skills. Soft skills are also valuable in this type of working environment such as teamwork, time management and adaptability. Self-discipline while working and doing the diary thesis will ensure to bring great and desirable results.

1.4 Objectives and Limits

The main objective of the diary thesis is to foster the trainee's professional growth through self-discipline and analysis of experiences over the designated timeframe. Daily reports will be logged and compiled into a weekly analysis throughout the 8-week period. This analysis will provide the trainee with an overview of work progress and offer feedback based on the obtained data. It will focus on identifying challenges and successes, and how this knowledge could be applied to any task. Knowing what the problems are, looking for solutions, and having a trial-and-error approach in addressing these problems will give vast data to the trainee how and what should be the proper way in dealing with them in the future. These weekly reports will continue until the end of the reporting timeframe which will be the basis of comparison, highlighting how the trainee demonstrates and applies these lessons.

Improving and supporting the professional development of the trainee will be the purpose of this diary thesis. The trainee's role during the observation period is to be flexible with changes. Participation with other projects and if time permits, the start of the actual Mendix-related project or other projects that are code or non-code related can be the changes that might happen over the course of writing the diary. Due to the non-disclosure agreement, the reporting will exclude any information that could compromise data security or reveal sensitive details about the companies involved, as well as any irrelevant information. In addition, the trainee's manager will be informed about the diary thesis and could ask for a grant that the trainee had not broken any security protocols, otherwise, future edits must be done.

1.5 Key Professional Concepts

Agile Development - Agile is a set of methods and methodologies that integrates project coordination and design, architecture, and process enhancement, addressing every phase of traditional software engineering.

Scrum - A simple and lightweight framework that helps teams deliver value by solving complex problems adaptively. It is an agile framework that guides teams in organizing and managing their work using defined values, principles, and practices (Scrum.org 2024).

Low-code - A visual developmental approach that accelerates application delivery with minimal coding. It provides tools like visual platforms, built-in connectors, code templates, benefitting both non-professional and professional developers (IBM Cloud Education 2022).

Mendix - A high-productivity development platform for building, deploying, and managing mobile and web apps which supports agility and seamless deployment on-premises or in the cloud. It is a low-code platform enabling easy app development and scalability throughout the process (Potanski 10 April 2024).

Mendix Studio Pro - Desktop IDE application designed for developers to build simple to complex applications, including custom logic, coding and integrations (Mendix 2024).

Domain model - Visual representation that illustrates the relationships, key concepts and behaviors of interconnected real-word objects or entities (Rodriguez 29 September 2021).

Microflow - Visual building blocks that add logic to Mendix applications. It represents business processes and enables workflow design with drag-and-drop features to automate tasks and improve functionality (Mendix Docs 2024a).

Entity – Represents a data object, defining its structure and properties, and is used to store and manage data in the application's database (Mendix Docs 2024b).

1.6 Overlay Matrix

The overlay matrix on the table below will be a representation of the trainee's reflective process during the diary thesis, mapping overall learning professional development themes and goals. The diary-based thesis is structured to promote professional growth in Scrum Methodology and Low-code program in using the Mendix platform, adopting a specific, measurable, achievable and atomic approach. Daily monitoring focus on documenting specific tasks and challenges encountered to monitor progress. Weekly reflections incorporate the analysis of relevant sources to provide alternative perspective and findings or support current knowledge that foster growth in Mendix expertise. The overlay matrix facilitates focused, time-bound progress that are aligned with the professional themes and goals regarding Mendix.

With the Overlay Matrix, Week 1 concentrated mainly on the Scrum Methodology as its development theme. It is to establish the crucial role of Scrum and Agile Development that plays in the low-code platform such as Mendix. Within the week, the trainee has better knowledge on different aspects such as on how to refine User Stories and understanding the Fundamentals of Scrum in general perspective which led to the outcome having Intermediate from a Basic point of view.

For the Low-code Program development theme, the trainee profoundly increased its knowledge and skills for the remaining of the diary monitoring. This focused on understanding about low-code

platform such as building reusable components of low-code which were then applied using the Mendix platform. This gave an out of the box experience for the trainee to have a general outlook on low-code regardless if it was used in Mendix or other platforms.

The Mendix Platform development theme has been the concentration of this diary thesis. The two formerly mentioned developmental themes, though function as main themes on their own, they also served as a support to this particular theme. This has covered various aspects of Mendix from understanding its basic fundamentals, proper use of Mendix Studio Pro IDE, to critically applying security configurations. The different objectives of the theme are spread throughout the weeks which gives various results or outcome for the trainee. This also reflected which field on Mendix Platform needs more developments, time and action.

The Overlay Matrix is specifically tailored to follow this sequence as it provides a structured and incremental approach on how the trainee should engage with the Mendix platform. Each step in the Overlay Matrix is designed to build upon the previous one, ensuring that the trainee develop a solid understanding of fundamental concepts before progressing to more advanced features. This careful progression helps avoid overwhelming the trainee, instead fosters confidence and competence at a manageable pace. By structuring the learning process this way, the Overlay Matrix ensures that the trainee is equipped with the necessary skills to effectively navigate and utilize the Mendix platform, leading to both short-term and long-term success.

Development Theme	**Development objective**	**Starting Point**	**Weekly analysis**	**Outcome**
Scrum Methodology	Understanding how Mendix implements Scrum	Basic Knowledge Basic Skills	Week 1	Intermediate Knowledge Intermediate Skills
Low-code Program	Understand what a Low-code platform is	Basic Knowledge Basic Skills	Week 2 - 8	Intermediate Knowledge Intermediate Skills
Mendix Platform	Mastering Basic Fundamentals of Mendix	Basic Knowledge Basic Skills	Week 1 - 8	Intermediate Knowledge Intermediate Skills
	Correct use of Mendix IDE,	Basic Knowledge	Week 2 - 8	Intermediate Knowledge

Development Theme	Development objective	Starting Point	Weekly analysis	Outcome
	Mendix Studio Pro	Basic Skills		Intermediate Skills
	Understanding how to create Domain Model	Basic Knowledge Basic Skills	Week 2 - 4, Week 6	Intermediate Knowledge Basic Skills
	Proper use of Microflow	Basic Knowledge Basic Skills	Week 2 - 6, Week 8	Intermediate Knowledge Basic Skills
	Configuration of Mendix Security settings	Basic Knowledge Basic Skills	Week 5, Week 6	Intermediate Knowledge Basic Skills

Table 1: Overlay Matrix

2 Description of The Initial Situation

This chapter will provide a description of the initial situation that analyses the current work, stakeholders, and workplace interaction situations.

2.1 Analysis of Current Work

The main task of the trainee is skill improvement and build competence with Mendix platform. Handled tasks would be proper utilization of the IDE, Mendix Studio Pro and its components, possible integration of the third-party applications available in Marketplace that are necessary to build requirements of an application, and incorporating the programming language, Java and JavaScript, in a more complex situation. The trainee's responsibility is also to apply the theory of Scrum and Agile methodologies in approaching its work. In addition, application design will be handled together with the User Experience and User Interface (UX/UI) Designer that is if a customer-related project arises. Database management wherein creating data models which includes defining entities, attributes and functions, testing the application for bugs and ensuring that it meets the specified requirements will also be included. Lastly, documentation of the development process is fundamental for future updates.

In order to cope up with work, it is necessary to have a working knowledge about Scrum and Agile development. Background and solid understanding of the IDE and its usage is essential. Even though Mendix mentioned that the platform could be used by both coders and non-coders, knowledge of some programming languages is of high value especially when specific features are to be integrated. Developers with a background in technical programming such as JavaScript and Java can utilize the software, Mendix Studio Pro, in combination with CSS to build, customize, and enhance an application.

Part of the learning academy that Mendix offers cover the topics of Scrum and Agile awareness. Even if the trainee has had prior knowledge about these matters, repetition and application is still essential to retain these types of information. These topics are part of the learning academy; hence the trainee aims to go through with them. The trainee, as mentioned before, already acquired certain certificates offered by Mendix namely Rapid Developer and Intermediate Developer Certification. Going through again on these topics as part of the 8-week course will ensure continuity of learning and retain crucial knowledge that can be applied during and even after the diary thesis.

In relation to Mendix platform, obtaining and completing the certifications offered in the course would be the trainee's good investment. Deepening the expertise in Mendix Studio Pro and other

tools offered by the platform will increase career opportunities. Participating in projects related to low-code, specifically but not limited to Mendix, will boost the knowledge and skills required to deliver quality solutions. Investing also in high-code or traditional code together with low-code is necessary. Since low-code platforms such as Mendix are designed to be enhanced with custom code, investing time in mastering both Mendix-specific tools and traditional programming languages will keep the trainee at the forefront of development.

Considering that low-code platforms often involve collaboration between technical and non-technical stakeholders, honing communication skills are crucial. This will bridge the gap between the developers and business teams. To complete tasks successfully and efficiently, it is essential to communicate and collaborate with the team especially with those that are directly involved. As a software developer, understanding when to seek assistance and consistently clarifying requirements are key aspects of effective communication throughout the development process.

Mendix platform is somehow new to Finland market and new to Tietoevry company as well. A team was created in which the trainee has been a part of it. The team was created for the sole purpose of presenting Proof of Concepts to the customer. At the same time, the trainee was tasked to learn about Mendix while work was in progress. In this given situation, with humble justification, the trainee is at a competence of a Novice. Guidance from people more knowledgeable of Mendix platform is still needed because of limited knowledge and experience. Even though certifications have been accomplished, application of the said knowledge must be done, and the best situation to achieve this would become part of a customer-related project. Gradual shifting from simple tasks to more challenging ones will be the target goal for professional development. With this diary thesis, the trainee seeks to improve, continue learning and develop the skills that are needed to attain the next step which is a skilled performer until the trainee becomes an experienced expert in Mendix. It may or may not happen over the course of 8 weeks, but the insights gained will elevate and prove valuable for the trainee’s career.

2.2 Stakeholders

The project team consists of the Software Developers, Designer, Project Manager, Business Consultants, Testers and System Analyst. These members would be the project’s internal stakeholders while the Client/Customer, Government Agencies and End Users would be the external stakeholders. This line of stakeholders will only be viable once a decision from the ongoing customer negotiation pushes through.

The trainee's primary focus of interest will be on the project's Development and Operations, Design, Testing, and Business Consultation, serving as the main line of communication with the client to ensure the delivery of the requested requirements.

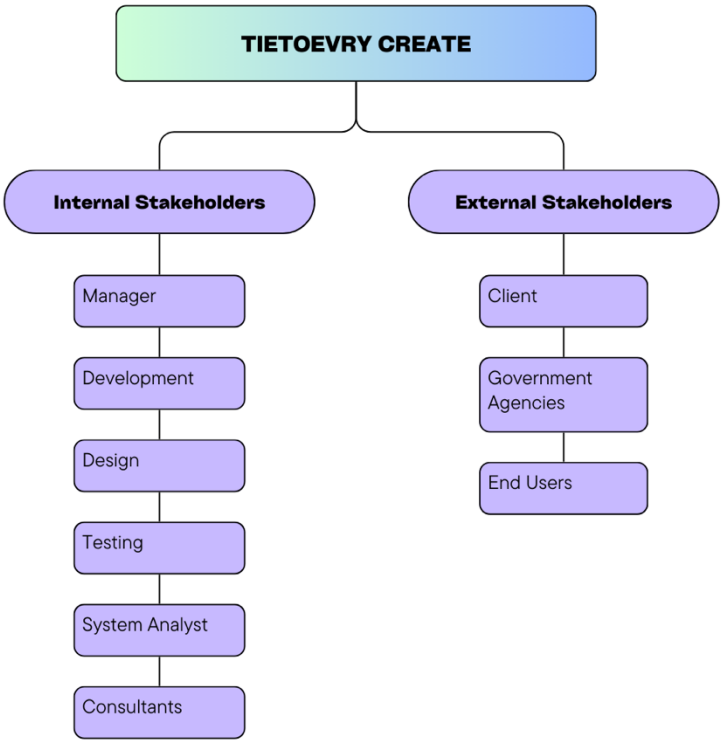

Figure 1: Internal and External Stakeholders

2.3 Interaction Situations

Having work-life balance has been quite the advantage of working in an IT company. Like most other IT companies, employees tend to work remotely and hold meetings through the use of different platforms. The trainee has had daily meetings as part of the Scrum methodology as well as weekly meeting with the team. Tasks that have been accomplished and tasks which are on-going have been discussed. The mode of communication of the team is mainly through Microsoft Teams and is in English language. However, when monthly and emergency meeting is to happen together with the client and other stakeholders, it is usually held in Finnish. Communication with the client is primarily handled through the project manager and business consultants and then relayed to the team.

Every after completion of PoC's, demonstration is presented to the client with all the stakeholders hopefully present. The trainee has been able to demonstrate in one of the PoC meetings but still requires more of communication skills in terms of the Finnish language.

Over the course of the thesis writing, communication will solely be held between the trainee and the trainee's colleagues. Challenges have always been present from the beginning since low-code is a new thing to the trainee as well as to the team. It requires a lot of time to learn and hopefully, be able to apply it. Since the team is in a self-learning phase, it is hard to identify and be on the same page as the team. Some could already be part of other projects, busy, some could not be reached, or some uncertainties could happen. Either way, being able to identify these hurdles, the trainee will be able to ask for assistance from the team whomever is available. One big problem for the trainee is if during the dairy thesis, the negotiation will still be on-going or that the project won't persist to exist, or the trainee might be involved in a project that is entirely different, or something else. In that context, knowledge and skills will still evolve throughout and even after the thesis is done and the trainee will still have the discipline and positive approach to these kinds of situations that will develop the trainee's professional growth.

3 Diary entries

The diary entries by the trainee will be followed throughout the 8-week observation period. This will reflect how the trainee works while making sure to seek professional development growth through the carefully thought developmental themes and goals.

Before starting the diary entries, the trainee was part of a small team that produced Proof of Concepts for a prospective client. During this time, the trainee was simultaneously learning about Mendix alongside the small team, with monthly internal meeting and updates on project progress and best practices on using Mendix. Contents and small projects developed using the Mendix platform were completed and shared internally. Some of the work done by the trainee during this period has been incorporated into the contents of this diary thesis.

The diary thesis serves as a letter by the trainee to himself and to others who are interested not only in Mendix but also in low-code development. It aims to develop the trainee's competence, skills, knowledge, work balance, and soft skills. Through the daily documentation, weekly analysis and reflection, the trainee captures personal growth, challenged faced, and insights gained, creating a valuable resource for future reference and for other who may want to embark on a similar learning journey.

3.1 Observation week 1

Monday 21 October 2024

Today's work focused on learning about Agile framework, specifically the Scrum methodology, as part of the professional development theme set by the trainee.

The day began with reading a module that lets the trainee be introduced with the objectives and provided information about the use case scenario. It continued with walking through the recommended training path, making notes and taking into consideration relevant knowledge that was also tied with the daily goal the trainee has set for the day which was understanding the fundamentals of Scrum. The task for today revolved around the Agile awareness, offering insights into how Agile functions within the Mendix platform. The concept of Digital Execution was explored including the stages and its core principles. Moreover, the 4P strategies of Digital Execution – People, Portfolio, Process, and Platforms were also covered, demonstrating how these principles relate to the application of Agile process. While there are a lot of agile

frameworks such as Kanban, Lean, and Extreme Programming, the Mendix Agile approach is rooted in the Scrum methodology.

Reflecting on today's task, the trainee gained crucial information about Agile and Scrum, building a strong foundational understanding. This gave the trainee solid insights into the stages of Digital Execution and the role of the 4P strategies in making the vision of an application a reality. Having this core knowledge gave the trainee the needed information to continue improving and enhancing what was already gained.

Tuesday 22 October 2024

Today's task was a continuation of the Agile framework as part of the theme, Scrum methodology. The task on hand was in relation to the Agile awareness but specifically how to have an Agile mindset, its values and principles. The learning goal for the day was continued understanding of the fundamentals of Scrum.

The trainee's work focused on understanding the values and principles of Agile, which concludes the foundation of Scrum. The day began with learning how Agile mindset works and how this would correlate with today's theme and goals. Exploring the Agile mindset, four values and twelve principles have been formulated from them which characterizes the Agile approach. Additionally, roles and responsibilities within a Mendix Scrum team were reviewed, including the Product Owner, Scrum master, Development Team, Subject Matter Experts (SMEs), Business Owners and Stakeholders. The trainee also tackled Scrum artifacts such as Product backlog, Sprint Backlog, and Product Increment. Concepts like Epics, User Stories, Tasks, and Definition of Done were covered, aligning with the trainee's goal of understanding how Mendix applies Scrum principles.

The trainee gained key knowledge about the Agile mindset, values, principles, and Scrum artifacts through repetition to retain information. Further learning helped the trainee to be reinforced with foundational knowledge of Agile, highlighting the need for continuous learning and growth in Scrum methodology.

Wednesday 23 October 2024

For today's work, the focused professional development theme was Scrum methodology, specifically the Scrum events. This will be part of yesterdays' work on understanding the fundamentals of Scrum and how Mendix applies the methodology.

The daily task focused on Scrum events which envelopes the trainee's set development goals. These events represent the process of each Sprint namely Sprint Planning, Sprint Review, Sprint Retrospective and Daily Stand-Up, as well as Product Backlog Refinement, which, may not be officially part of Scrum event, is still essential. The trainee gained valuable insights and reinforced its understanding of these events on what needs to be done within each of them. Fibonacci-like sequence was also touched in this learning on how Mendix approaches its user stories in estimating them and the factors involved in completing them, such as Definition of Done and the Acceptance Criteria. The trainee also gained a clearer understanding of the Scrum Masters responsibilities through the study of Sprint Planning, Review, and Retrospective events.

This knowledge certainly contributed with the development growth of the trainee by knowing how each of these event work in balance. The challenge with this was the application of these learned knowledge but nonetheless, it will be applied when the time is right. Today's work made it possible for the trainee to enhance its core understanding of Scrum and its importance in being proficient enough in the future.

Thursday 24 October 2024

Today's work was a continuation of knowledge about Scrum, how Mendix implements Scrum and therefore it also includes understanding the basic fundamentals of Mendix. Writing user stories was also included as a challenge.

The daily task focused on applying Scrum Artifacts and understanding how the Product Owner interact with the Business Owners and Stakeholders. Understanding how Mendix approaches and how it implements Scrum, additional steps were known to ensure the validity and basic goals of the product. This includes three steps: Business Case, form a Cross-functional Team, and defining a Minimal Viable Product (MVP). The trainee gained insights from these in understanding whether an app idea would be suitable for Mendix and whether a team would be capable in collaborating through an Agile approach. While the concept of MVP was familiar, the trainee found these steps new and challenging. The task also involved applying Scrum events in the Mendix environment by creating a blank application. In addition, proper writing of user stories was observed.

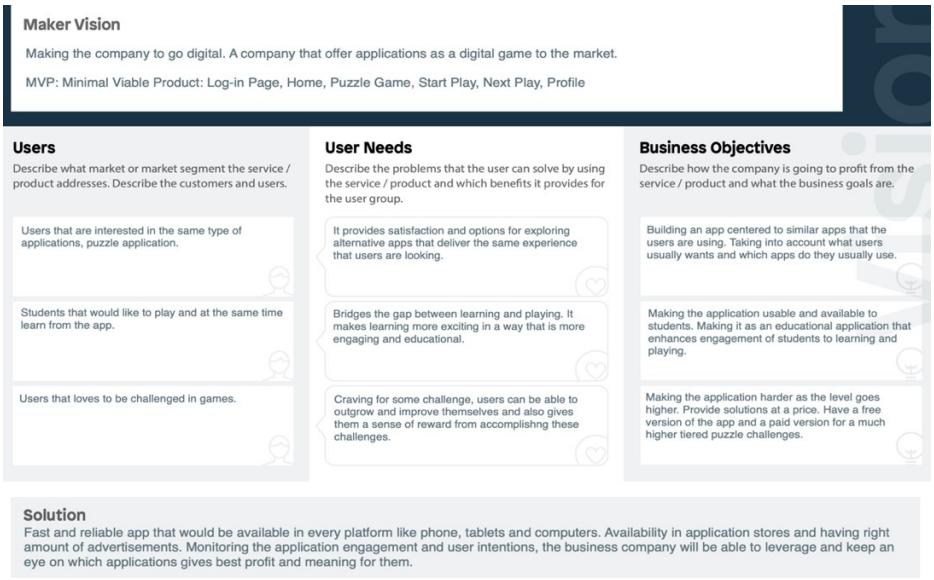

Figure 2. Collaboration Process Business Case using Mendix Template.

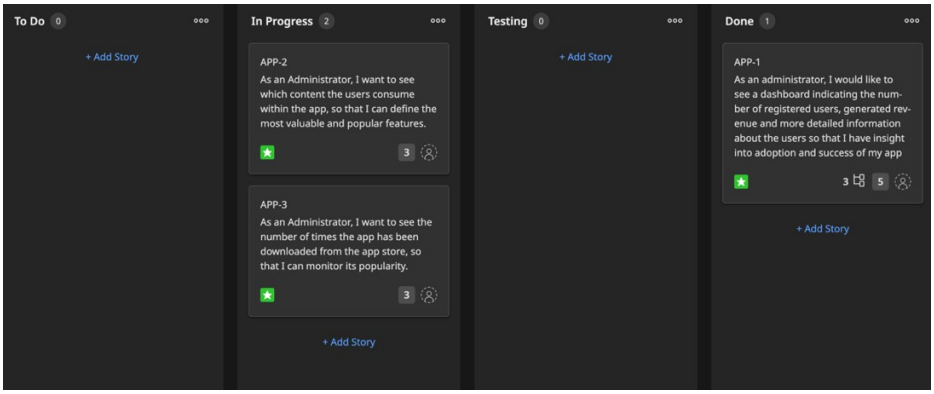

Figure 3. Sample Sprint with User Stories using Mendix Platform.

Today's task advanced the trainee's development by applying knowledge of Mendix's Scrum events, providing valuable insights. While user stories could be improved, the application of Scrum events was successful, enhancing the trainee's understanding through practice and learning.

Friday 25 October 2024

Today's task focused on Scrum methodology and the Mendix Platform. The trainee continued learning the fundamentals of Scrum, including user stories and Mendix's Scrum implementation, while also covering the responsibilities of a Scrum Master.

Today's work focused on understanding the value of business case and knowing if an idea would fit into Mendix definitions. The trainee learned to improves its knowledge in assessing

an app's business focus, including target users, implementation channel, functions, and integrations. The importance of team values—Focus, Courage, Openness, Commitment, and Respect—was also highlighted, emphasizing their role in achieving collective outcomes. The trainee applied these values to create a solid product backlog, epics, user stories, and product increments. Additionally, the trainee completed a Persona template and Wireframe and learned Mendix's specific approach to writing user stories. Moreover, understanding the Scrum Master's responsibilities were part of the trainee's learning. Meeting facilitation, determining the right form and ensuring the right people are involve were some of the learning the trainee has obtained.

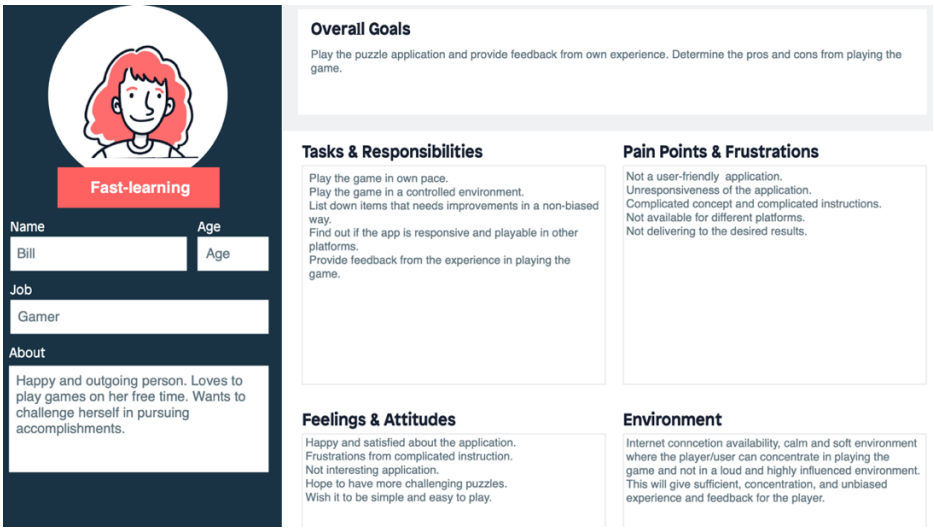

Figure 4. Persona based on use case using Mendix template.

Week 1 Analysis: Scrum and Agile

The trainee's 1st weekly analysis was directly influenced by the week's diary monitoring. The topic for the weekly analysis focuses on Scrum, more specifically about Scrum Artifacts. The Product Backlog, Sprint Backlog and Product Increment are the key elements belonging to it. This analysis will be a learning type of analysis about Scrum and how Mendix implements its view on this methodology.

Scrum is a straightforward framework that can be implemented directly to assess whether its philosophy, theory, and structure effectively support goal achievement and value creation. By applying Scrum as intended, teams can evaluate its impact on productivity, collaboration, and delivering tangible results (Schwaber and Sutherland 2020). Part of the trainee's learning is going through Agile awareness, hence learning about Scrum. The trainee initially

believed that understanding Scrum was simply a matter of forming a team, defining roles, and familiarizing themselves with the values, events and artifacts but it was actually more than that. With the knowledge gained by the trainee over the week, it became clear that it had developed better understanding of what Scrum truly is and how it can work to one's advantage.

Understanding Scrum being as simple as it is, surely did changed and evolved from the trainee's perspective. This was evident from what the trainee has read, and knowledge gained. Scrum principles for project success and its values, events and artifacts were further developed in the trainee's understanding as reflected through reading sources, learning and applying them whenever possible during the diary monitoring.

Scrum's artifacts represent both the work completed and the value created throughout the project. They are specifically designed to ensure transparency of essential information, allowing everyone involved to have a clear, shared understanding, which provides a solid foundation for making informed adaptations as needed (Schwaber and Sutherland 2020). This represents a collective work of items for the team to work on and execute it in a way that emphasizes its value and importance. These artifacts were applied in various stages of the Scrum process which were available to the Scrum team. For the trainee's weekly analysis, it focused on these Scrum artifacts as it piqued its interest and knowing more about them will serve as reinforcement of ideas.

The first artifact is the Product Backlog. The Product Backlog is an evolving, prioritized list of items required to enhance the product. It serves as the single, definitive source of work for the Scrum Team, guiding their efforts and ensuring that the most important tasks are addressed first to drive continuous improvement (Schwaber and Sutherland 2020). Learning throughout the monitoring week, the trainee has better understanding on what these artifacts are. Each artifact has its own commitment as it provided information about them. Product Backlog has Product Goal as its commitment which makes it as the target plan by the Scrum Team. Going through the weekly monitoring, the trainee had only limited information about Product Backlog let alone applying them in a weekly or in a day-to-day situation. According to Sharon Blanchard, it is common practice to write Product Backlog Items (PBIs) in User Story format. Many Scrum Teams find it beneficial to allocate a short amount of time during each Sprint to focus on refining and writing these user stories, ensuring that the backlog remains clear, well-defined, and ready for future work (s.a., subchapter Scrum Artifacts). Based on the trainee experience, epics, user stories and tasks were tackled and with reading through sources, it came to a clear understanding how crucial these user stories are in

affecting the success and velocity of the project. Writing user stories as part of the challenge were done but required more precision and practise. The same is with the Product Backlog Refinement. While Backlog Refinement is not a required event, it is a required activity. Most Scrum. As a reflection, the trainee only had prior knowledge about backlog refinement but had not usually applied them, but by reading and understanding them through these sources and learning paths, the trainee recognizes these as part of growing and enhancing its knowledge.

The second artifact is the Sprint Backlog. The Sprint Backlog consists of three key components: the Sprint Goal (the "why"), which defines the purpose of the Sprint; comprises the Product Backlog chosen for the Sprint (the "what"), which outlines the work to be completed; and an actionable plan for delivering the increment (the "how"), which details the steps the team will take to achieve the Sprint Goal and complete the selected work (Schwaber and Sutherland 2020). This happens during Sprint as the team identifies these items from the Product Backlog. As reflected from the diary monitoring, the trainee has applied these principles in writing user stories and placing them on a Scrum Board using the Mendix platform. This is planned and used by the Scrum Team as to have a centralized place for it to be accessible for everyone. Using the Scrum Board, the Sprint Backlog could be updated at ease throughout the Sprint. The information from these Sprint Backlog gives enough details as to how it could be presented and divided into much lower tasks that are feasible for the current Sprint. Knowing the Sprint Goal, which is the commitment tied to Sprint Backlog, the Scrum Team would be able to work together which creates a unified momentum and purpose to deliver and achieve the teams desired outcome. Looking up and reading through sources, the trainee had more understanding about Sprint Burndown Chart. This has only been lightly touched from the diary monitoring, though not as relevant as the others, it still gives great amount of information.

The third artifact is the Product Increment. An Increment is a tangible milestone that moves the team closer to achieving the Product Goal. Each Increment builds upon the previous ones, with thorough verification to ensure all Increments function cohesively. To provide value, the Increment must be fully usable, meaning it meets the necessary quality standards and can be implemented or delivered (Schwaber and Sutherland 2020). The trainee has had little information about these compared to the first two artifacts. The information gathered from the diary and searching through sources was deemed important. Knowing that the Sprint composed of user stories, these user stories that have been completed represents a Product Increment. As Scrum is an iterative process, Product Increments continuously evolved and changes too and at the end, the Product Increment will represent the product

which will be reviewed and presented to the customer. This only happens if the Product Increment is accepted within the criteria of the Definition of Done. The commitment associated with this artifact is the Definition of Done. With the experience of the trainee, Definition of Done is only known with theories and not through application. These are the items that the Product Increment should satisfy in order to be considered done.

Scrum Teams may also incorporate brainstorming activities as part of their collaborative efforts. These activities can range from simple to more structured sessions, with the primary goal of fostering creativity, problem-solving, generating idea (Schwaber and Sutherland 2020). Team collaboration is essential for delivering requested functionalities and meeting quality standards. Through regular communication and feedback, team members leverage diverse skills to solve problems and ensure product meets the needed deliveries. This collective effort promotes efficiency, accountability, and continuous improvements, leading towards the success of a project.

The professional development themes and goals addressed in this diary were highly valuable for the trainee's professional growth. The obtained knowledge and understanding from work then correlating them with the daily development themes and goals provided a clearer path as to how the trainee should approach each of them. The process of connecting practical work and theoretical framework provided deeper insights that contributed to the existing knowledge of the trainee which in turn provided new perspective.

Reflecting on this week's diary monitoring, the trainee found that these insights enriched its learning skills and broadened its understanding of effective work practices. Through application of the concepts learned highlighted areas for improvement, helping the trainee to identify strengths and weaknesses in approaching these matters. This allowed the trainee to develop a more structured mindset when tackling daily challenges, fostering a more proactive approach for developmental growth. The outcome of this week's diary has had significant impact on the trainee's development. It has laid a strong foundation for ongoing growth, as to how it approaches future tasks with clarity and improved understanding.

3.2 Observation week 2

Monday 28 October 2024

Today's work focused on Mendix Platform, particularly related to mastering basic fundamentals of Mendix, proper use of Domain Model through the Mendix Studio Pro software.

The daily task centred around the creation of a template, customization of pages and Domain Model with the use of the software, Mendix Studio Pro. Efforts began with the proper implementation of the template and understanding of the software interface. Creation of a page was then designed at the top navigation bar of the application. The navigation bar has different events, but with this particular task, the function was to ‘Show a page’. Within this page, the trainee’s task is to add a widget button that shows another page as a ‘Pop-up’ style when clicked. These pages were then connected to the Entities in the Domain Model to use it as its data source. With each Entity, items known as attributes were added, updated, or deleted according to the requirements of the created pages. To enhance maintainability, proper attribute naming, associations, and page naming conventions were observed. The core logic for how the flow and implementation of the pages was established, and the connection between these pages to the Domain Model was further understood, providing a foundation for future references.

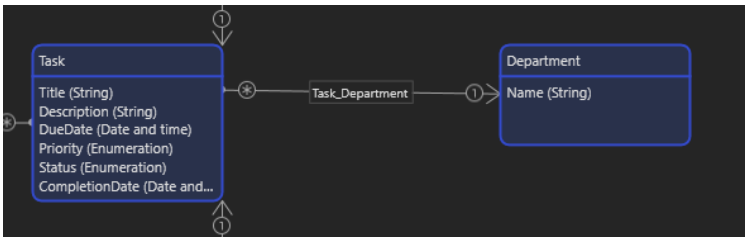

Figure 5. Domain Model with Task and Department Entities.

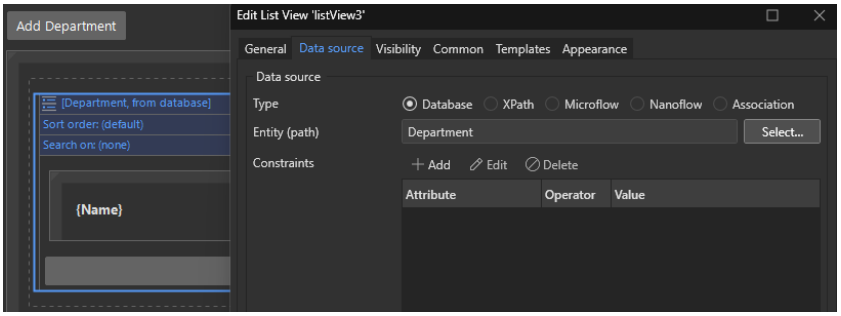

Figure 6. A page linked to the Entity Department used as a data source.

Reflecting on today’s work, the tasks done by the trainee strengthened its understanding of the Domain Model, how it works and how it was crucial for the pages created. Proper use of the Mendix Studio Pro was established and enhanced. Some challenges faced by the trainee were handling errors. Great thing with Mendix, the software itself provides ample information of what these errors were. Going through the given information, the trainee was able to navigate and solve them. Knowing these errors, it provided relevant information how the trainee would face them again if ever it arises. Nonetheless, further strategies will be used in order to

avoid and address them. Overall, the task of the trainee today highlighted the importance of continuous learning and exploration in mastering the Mendix Platform.

Tuesday 29 October 2024

Today's work focused on using Mendix Studio Pro as a continuation of professional growth to the trainee, particularly on using a Microflow in setting a value for an attribute in the Domain Model.

The task starts with creating an Enumeration attribute on the Domain Model that has values of either Achieved or Missed. The primary learning goal was to gain proficiency in connecting these values to the pages using a Microflow. Understanding how Microflow works, and its different activities revealed successful implementation of its function. Specifically, improvements were made at the Microflow template extending it with the use of activities called Decision and Change Object. Decision was used to validate if the certain conditions were met, in this case, whether the page has a due date and a completion date. Change Object then changes the status of the Enumeration attribute whether it was Achieved or Missed. The Microflow reflects the information through a sense of true or false flow.

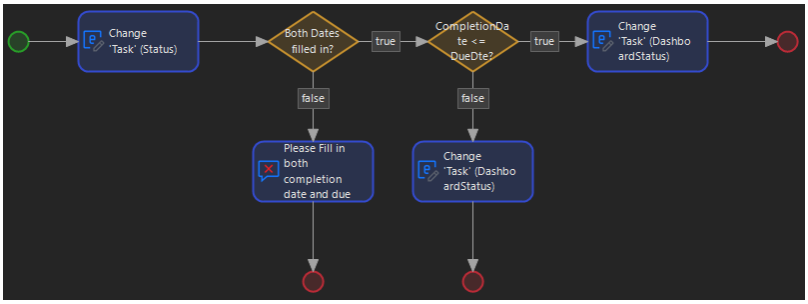

Figure 7. Microflow using Decision (Yellow) and Change Object (Blue) activities.

Reflecting on today's task, the trainee had an essential understanding of how the Microflow works. Continued learning about the Domain Model and using its different activities provided crucial information for the trainee to better understand them. Use of these activities required more application for the trainee to be familiarized with their use. The concepts, logic and principles behind these connections of Microflow, pages and Domain Model showed different outlook on how to deliver a product at a more flexible and faster way.

Wednesday 30 October 2024

Today's work involves using of an existing Excel spreadsheet and integrating it with the Mendix Platform. The development goal for the task is to create a Domain Model which reflects the contents of the spreadsheet then showing it to a page in the application.

The trainee's task began with having an existing Excel spreadsheet containing different kinds of information and then consuming it through Mendix Studio Pro to show its contents. With the use of Mendix, importing the spreadsheet made it easier and faster. The creation of the Entities in the Data Model was automatically generated by Mendix which reflects its capabilities in the field of low-code programming. The trainee then utilized the use of a widget called Data Grid to show the list of contents of the spreadsheet. Connecting the Data Grid's data source to the Entities made it possible to automatically fill and show the spreadsheet's contents.

Reflecting on today's work, the trainee gained insightful way to deal with spreadsheets and utilize Mendix platform to show its contents. This implies valuable knowledge for the trainee when working with importing existing spreadsheets but nonetheless required more practice when dealing with larger and more demanding data. Through this experience, the trainee was able to properly display the data that reflected the Excel spreadsheet.

Thursday 31 October 2024

Today's task was related to creating pages and using different widgets to build their components. Moreover, the task reflected the developmental goal of the trainee in regard to understanding low-code through the use of the Mendix Studio Pro.

The task was initiated with a template provided by Mendix as the starting point. Understanding the widgets and their uses provided the needed information as to how and why they were used. The homepage was then built using a widget called Layout Grid as its main frame. Since the layout grid came in with a pre-installed column, duplicating was done to create four columns. Four more pages were then created and then linked to the four columns through a button widget. Errors arose from these pages since data sources were missing. Part of the trainee's task was to understand the best practices of using the Mendix Studio Pro. Committing the changes made on the application with unsolved errors is possible but not recommended. Still committing and saving the project with clear description of what was done even with errors were part of the learning curve for the trainee.

The work reflected the continued work of the trainee in using Mendix Studio Pro, especially in creating pages and the use of widgets. Learning from these errors gave better understanding of what the next steps would be. Overall, the trainee had achieved more velocity on creating

these pages and gained understanding on the concepts of low-code in delivering faster products.

Friday 1 November 2024

Today's task centred around the pages from yesterday's task and creating a Domain Model that represents each of them. The developmental goal for the trainee focused on utilizing what was learned so far to further improve the skills and knowledge in low-code and using the Mendix Studio Pro software.

The concept for today circled around solving errors and connecting these pages to its data sources. Four entities with their own attributes were then created to represent the four pages. Errors were then solved by linking each page to its designated entity. Contents of the page were then showed with the use of dynamic data, data that could be changed by the user. These dynamic data were created through a button widget which opens up another page which enables the user to populate them. Additionally, navigation menu for each of the pages were created in order to be accessible anywhere on the application. Changes were then committed, this time without any errors.

With today's work, the trainee successfully solved the errors and able to link the pages with the Domain Model. Complicated logic was faced by the trainee in the fields of populating the pages with data but managed to overcome them. Furthermore, learning and understanding the concepts of low-code provided value to the trainee.

Weekly 2 Analysis: Mendix and Low-Code

This week's analysis concentrated on exploring Mendix, a powerful low-code platform that allows one to create solutions with little to no coding. It also provides the ability to design and deploy applications at a much faster and easier pace. This analysis adopts a learning-oriented approach, going deeper into the principles of low-code development and highlighting Mendix as a leading platform in this space. It aims to provide valuable insights into these concepts and serve as a resource for the trainee and to anyone seeking to understand more about low-code development and the Mendix environment.

As reflected with the daily monitoring done by the trainee, it focused on further learning, understanding and application of knowledge which carved the foundations of this week's analysis. What the trainee gained from using the Mendix IDE, Mendix Studio Pro, was vital for the

weeks to come for it is the centre of focus of its work. The knowledge learned starting from the use of templates and applying modifications and alterations to pages then using different components of it up to committing them to the server provided vital flow of how it affects the understanding of the trainee in regard to low-code and Mendix.

The concept of low-code has been around for several years, offering an alternative approach to application development. This approach enables faster development, delivery, and production of applications by minimizing the reliability on extensive coding. Often associated with the term "citizen developer," it empowers business users with little to no traditional programming experience to independently create complete applications (Kenneweg, Kasam & McMullen 2021, chapter 1). Low-code is tied with the use of visual driven tools and technologies, such as Mendix, which reduces the need for traditional coding. By adopting a drag-and-drop structure, it makes it easier for users as well as the trainee to bypass the time -consuming task of manual building of code-related work. This approach accelerates production and enhances overall progress.

With the concept of low-code in Mendix, the visual presentation of things made it easier for the trainee to follow what has been done and what needs to be done. This provided a visual flow of how the application would look like without even being published. The use of the Mendix Studio Pro required understanding and learning as part of developmental growth of the trainee. Working while studying the use of the software is a learning curve for the trainee to adapt and acquire the needed skills and knowledge to improve and apply its full potential for a future career that low-code could possibly provide.

Mendix is a leading low-code platform that offers tools to build, test, and deploy applications efficiently. As one of the most popular low-code platforms on the market, it enables the development of both mobile and web applications while adhering to agile best practices, ensuring high productivity and flexibility (Kenneweg, Kasam & McMullen 2021, chapter 1). As already noted, it allows faster creation of applications therefore save a lot of resources from businesses. It is a great choice for businesses who needs digital transformation. According to Gosiewski and Swiatek, Mendix is one of the world’s leading low-code platforms, renowned for its powerful and comprehensive features, supported by a robust team and an active, engaged community (2024).

The use of Mendix Studio Pro from the daily monitoring reflected how low-code and Mendix correlates with each other. Part of the trainee’s tasks were creating pages while facilitating the drag-and-drop capability of the software. Building an application without any coding is made possible by the drag-and-drop functionality of low-code, allowing users to easily place

components onto a canvas or template. While anyone can utilize this feature, achieving advanced functionality with a low-code tool often requires a deeper understanding of concepts like Excel formulas and spreadsheet (Bloomberg 2020, 22-24). As reflected from the trainee's work related to templates and spreadsheets, implementation of both activities was successful though it still required further skills when dealing with a more complex structure. The drag-and-drop feature were well demonstrated as part of the pages that were built. The pages' components were also built with this type of feature which can be easily understood with the low-code platforms visual presentation of the application. Mendix Studio Pro enables users to design their applications by creating pages, adding logic, and configuring settings. This optimizes process, supports testing and deployment across various environments. (Kenneweg, Kasam & McMullen 2021, chapter 4).

Understanding the Domain Model was part of the trainee's continuous learning. This is a big part of work as this is where the core of information is stored and facilitated throughout the software. The Domain Model in a Mendix app represents the application's data layer. Mendix Studio Pro offers all the necessary tools to design the app's data structure and present that data to end users within the application (Kenneweg, Kasam & McMullen 2021, chapter 6). The trainee was able to demonstrate the learned skills and knowledge of the Domain Model from the daily monitoring and yet, it still has a lot to offer in terms of skill build up and implementing its best practices. While working with the Domain Model, the trainee has crossed path with its concepts that proved to be of importance to the overall structure of Mendix. The Domain Model in Mendix allows one to visually design the app's data architecture. It consists of entities (objects) similar to tables in a relational database, with attributes (columns) to store data and associations to show how data objects relate to each other (Kenneweg, Kasam & McMullen 2021, chapter 6).

Working with attributes, it is understood that it describes an entity. As an example, the Member entity will include attributes that correspond to real-life characteristics, such as a name. This could be represented by a string-type attribute, like the "FullName" attribute (Kenneweg, Kasam & McMullen 2021, chapter 6). While working with the application and deep diving on these relevant sources, the trainee has had better understanding on how to differently approach its work more efficiently. Such as when working with entities and pages, properties of the entitles and pages could be manipulated from either the properties window or from the Properties panel in the Studio Po.

It is important to note that attributes and entity names are written in PascalCase. Following a standard naming convention for attributes, entities, and other elements in an application is

highly recommended for consistency and clarity (Kenneweg, Kasam & McMullen 2021, chapter 6). With this learning, the trainee was able to utilize the use of Mendix Studio Pro in a faster and easier way. Proper naming of entities was obtained, and errors were avoided as well while keeping them in mind. Knowing this, the trainee understood that naming the elements in PascalCase enhances readability and collaboration among developers as it also follows the practices of Mendix platform.

The experiences that happened to the trainee with the daily monitoring of work and events were a stepping stone for further development and achievement of the trainee's goals. The understanding of low-code was reflected with the trainee's use of the Mendix platform in implementing the key concepts of low-code. With the concept of drag-and-drop, it reinforced the theories and sources that the trainee gathered and read. The ease of using this drag-and-drop concept without a doubt made the development of the application faster without the use of heavy hand coding. The errors that arose during the structure of the pages were shown in a more clear and informative way. Through this, the trainee was able to witness the concept of low-code in addressing them effectively. The sources supported these concepts by providing examples as to how to effectively address them in other ways and approach them in alternative methods. Mendix, as a low-code platform, supports its approach by offering a wide area of developmental tools. Visual development concepts of the sources were reflected on the trainee's work as part of the low-code power of Mendix. While building the components of the pages, visual representation was present, and it gave real-time updates to the trainee how the app would look like.

The knowledge and insights obtained by the trainee with the daily monitoring were utilized properly as demonstrated with the effective approach of the trainee in completing its daily tasks and effectively approaching the challenges that was faced during the week. The trainee became aware of its strengths and weaknesses and used them to its advantage in order to better understand and know more how to address the incoming week's tasks. The expertise that was gained described the different concepts and techniques that the trainee learned about the concept of low-code and Mendix. Using the Mendix Studio Pro software, the trainee further increased its knowledge in utilizing it better as to lessen the workload and finish the task even faster.

Part of the trainee's overall goal is empowering its developmental growth. This includes self-development. With the wisdom obtained last week and through gathering of relevant sources, the trainee has improved its understanding of low-code and Mendix. Through harnessing the gathered knowledge in using the Mendix Studio Pro, improved and more

effective approach has been obtained. Self-development had surely taken place but there is still a long way of learning opportunities for the trainee to go through. Skills has been acquired and developed as demonstrated with the page creation and error handling. The principles of low-code were further developed. Time management was better observed by the trainee for it experienced less burned out compared to last week's analysis. Professional developmental themes and goals were kept in mind and were aligned with the daily tasks.

3.3 Observation week 3

Monday 4 November 2024

Today's work focused on Mendix as part of the continuation of last week's work, particularly related to Domain Model's entity and nesting its data.

The daily task concerned about developing a page that displays dynamic data though the use of Data view and List view widgets. Efforts began by using a blank template that was used as an overview page and placing the widgets with the process of nesting, filtering in the data widgets. Data view widget was implemented to the page and inside it, a List view was placed in order to display only a specific type of data or information. The data displayed on the page was possible through manipulation of an entity that was connecting two different entities.

Efforts were reflected by the trainee with proper implementation of data through nesting. Through recognizing the use of nested data, the trainee was able to demonstrate when to use them. Errors were faced during its implementation, though it was resolved it still required further application or practice. Selecting the correct entity from the Domain Model imposes showing the right information or data that are reflected onto the page.

Tuesday 5 November 2024

Today's work centred on Mendix specifically focusing on configuration of Microflow. Modelling a Microflow to behave accordingly and complete a task just like a button widget would in creating an object and go to a page.

The task began with the trainee creating a button widget at a page which eventually when clicked, opened a scheduling page. The button widget was then set to call a microflow instead of just showing a page as its events. Reason for this was to have a pre-filled or pre-selected option readily available based on what was clicked. Two activities were then added

- Create Object and Show page onto the Microflow. Errors were faced by the trainee that are linked to configuration of these activities in the Microflow. Choosing the correct entity and linking it to the Create object activity solved the error. Selecting the right Member, property of an entity which includes both attributes and associations, and setting its value is where the low-code side of Mendix applies. This created the Variable, placeholder within the microflow, of the entity. Configuring the Show a page activity with the right parameter was done by selecting the variable created by the Create object.

Reflecting on today's activity, the trainee improved the skills towards Microflows through learning its configurations. Understanding the logic behind the microflow was informative on how the microflow runs. With the configurations done, the trainee was able to show the selected page with a pre-filled data that was required. This reflected the gained skill of the trainee on applying alternative ways in displaying data.

Wednesday 6 November 2024

Today's work focused on Mendix Microflow, more specifically the use of an expression called Microflow Expression to calculate a certain requirement and validate certain conditions.

The task for today faced was to calculate the EndDate based on the given StartDate and its Duration. This was a continuation of yesterday's work related to scheduling. The task was in relation to the Change Object activity wherein EndDate has been chosen as a 'Member' and configuring its value using Microflow expressions. The expression used by the trainee to calculate the EndDate was a function called addDays(). The function expected to have its argument types which the trainee addressed by calling the StartDate and the Duration from the entities.

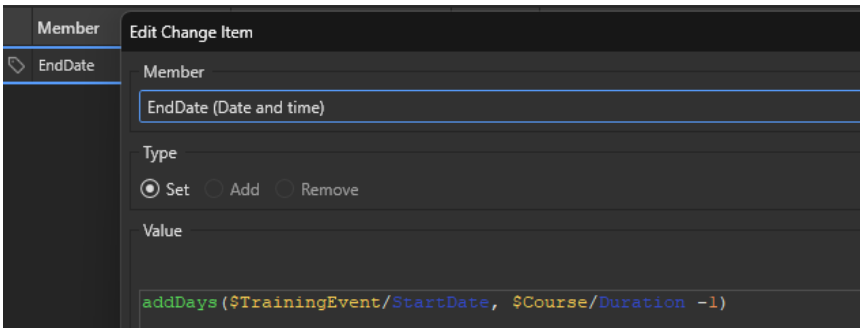

Figure 8. Calculating the EndDate using the addDays() function.

The work reflected how the trainee handled Microflow Expressions and how to effectively identify and use them correctly. This still needs further skill development as there are tons of

Microflow Expressions available which addresses different problems. Using the addDays() function came with potential errors description shown on the Error pane. This gave the trainee time to potentially resolve them while working on them at the same time.

Thursday 7 November 2024

Today's work focused on valid data on Domain Model and Microflows. Validations rules that are set to give information whether an input field is required, unique or equals to, are among the rules.

The task began with the trainee determining which input field on the pages needs to have validation rules and what type or types of validation rules needs to be applied. First validation rules were applied to the Domain Model. This was done as general rules solely for the entities and its attributes. The trainee applied the Required, Unique and Range validation rules accordingly and have validation feedback written if even a user failed to comply with these rules. The second validation rules were done onto another page's Microflow which is called by a Save button widget. The same validation rule types were applied accordingly for these input fields on the page. Changes done were then committed to the team server with the appropriate description.

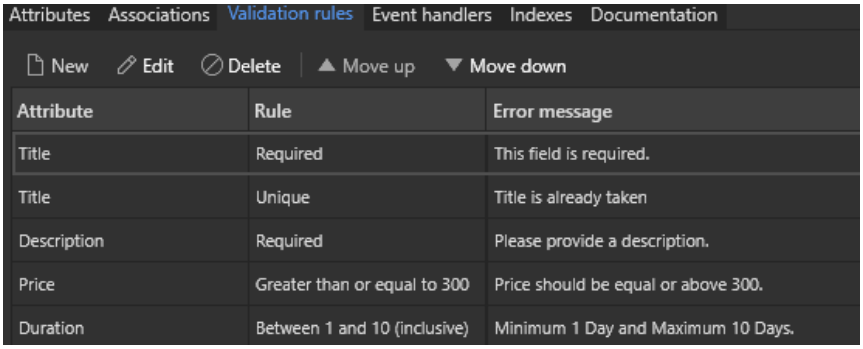

Figure 9. Validation rules on a Domain Model.

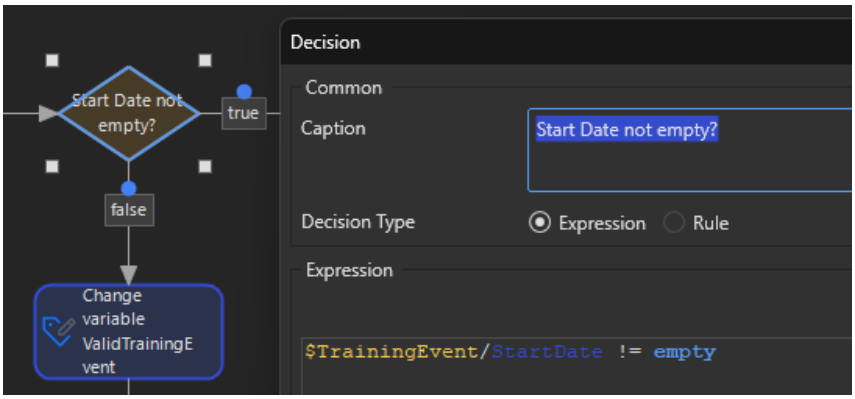

Figure 10. Validation rule on a Microflow.

Today's task reflects the trainee's skill in knowing validation rules. Knowing which types and rules to be applied, and where to put them, whether to a Domain Model or Microflow, displayed the trainee's application of learned knowledge. Though it was not quite extensive, it still was considered to be an important aspect for the trainee's professional growth.

Friday 8 November 2024

Today's work was related to deletion of data or objects on pages. This was a continuation of the work about Entities and its associations. Association refers to how two entities are connected to each other.

The concept for today was setting properly the delete function of associations, whether or not delete only an entity's objects once a delete button widget is activated or delete only an object if it is not associated with another object. The task began with adding a delete button widget onto pages where a list of data was displayed. Both of these delete behaviours were applied to separate pages with different list of data displayed. Error feedback was created as part of best practices prompting an error if ever an object is to be deleted while it is still associated with another object.

Today's task contributed to the trainee's development goals through the skills and knowledge acquired. Challenges faced by the trainee, however, was choosing the right delete behaviour. It was confronted with a trial-and-error approach to solve this problem. Deletion of data were still to be improved in terms of skills and adherence to its rules to avoid unintended loss or mishandling of information.

Week 3 Analysis: Process Automation

This week's analysis concentrated on Microflow concept of Mendix. This analysis further covered how Mendix approached the process automation through the use of Microflows. The weekly analysis was about the professional development theme, Mendix Platform, and also as part of developmental goals in mastering the fundamentals of Mendix and efficient use of Microflow. The analysis aims to reinforce the skills and knowledge that was reflected from the daily monitoring of the trainee. Not only the analysis provides information about Microflow and Process Automation, but it also provides insights on applying these concepts.

Through daily monitoring of work-related tasks, the knowledge gained from them reflected the trainee's understanding Mendix in general and utilizing the use of Mendix Studio Pro software. This was further demonstrated when the trainee worked with the implementation of pages and configured them through the Domain Model and applied the process of automation through the use of Microflows.

According to Salesforce, process automation uses software to automate business functions, helping organizations achieve goals like product production, employee onboarding, or customer service. It enhances efficiency and agility by enabling the creation of workflows and task automation with minimal coding (2023). This is the case of Mendix in its approach to business processes in terms of low-code development. In a nutshell, Mendix automate most of the tasks in the development process that was also seen from the trainee's implementation of Microflows while working on the application. It was further explained by SAP that business process management focused on improving back-end productivity by automating repetitive tasks. It involves using software and technology to automate business functions and achieve organizational goals, like production or employee onboarding. With advancements in AI and machine learning, modern process automations' scope and scalability are nearly limitless, driving digital transformation. (2024).

In Mendix, a Microflow visually represents program logic through diagrams and activities, enabling the automation of processes like retrieving, modifying, and manipulating data. It represents an applications' logic to model workflows. It also controls user interactions by displaying pages and guiding the flow of information, all while simplifying development through its interactive interface. (Kenneweg, Kasam & McMullen 2021, chapter 8). The application of this knowledge by the trainee was observed during the weekly monitoring. Modifications and retrieving of data from the Domain Model were executed by the trainee which were then shown to the pages on the application.

Working with Microflow, the trainee gained core knowledge of the basic events of the Microflow and how the logic of it works. A well-designed Microflow follows a clear, left-to-right flow that represents the "happy path," where processes proceed as expected. When negative decisions or exceptions occur, the flow branches down to specific decision points, where conditions determine the next steps. This structure ensures that the logic is easy to follow, allowing for effective handling of different outcomes and maintaining the flow's clarity and simplicity (Kenneweg, Kasam & McMullen 2021, chapter 8). This was further elaborated from the daily monitoring of the trainee wherein it follows a path like true or false until it reaches an end point.

Decision activities were used by the trainee on these Microflows (e.g., Thursday's entry) to validate a certain attribute that it is not empty. The logic of the Microflow follows a specific path, depending on whether the requirement is met. If the condition is false, a validation feedback message is triggered, and the object is not committed. However, if the condition is true, the Microflow continues along the happy path and commits the object (Kenneweg, Kasam & McMullen 2021, chapter 8).

The Validation Feedback activity performs a validation check on the input or data. If the check fails, it displays a message below the corresponding widget, highlighting the specific attribute or association that did not pass the validation. This alert serves to inform the user of the issue, enabling them to correct the input or data before proceeding (Mendix Docs 2024c). Validation feedback was used by the trainee in providing information to the user as to what the error would be. The process automation is shown here with the Microflows ability to check every object with only a single click rather than checking them one by one.

Input widgets can initiate a Microflow that automatically fills in data based on user input or predefined rules. Additionally, expressions provide a way to write custom logic, allowing one to create complex functions, apply validations, and modify values dynamically within the application. This combination of triggers and expressions enhances the flexibility and functionality of the app, enabling seamless data handling and user interactions (Akondi 28 August 2020). Automation of process was observed at the pages that were manipulated by the trainee. This was the implementation of input fields. With the use of input fields, originally it only can be filled with manual user input, but through the use of Microflow, automatic population of those input fields were done.

DateTime functions enable interaction with DateTime data, often used for tasks such as calculating the number of days between dates or performing various date-related operations. These functions can be used for comparing dates, subtracting days, or manipulating date values in other ways, allowing for precise control over time-based calculations and comparisons within an application (Kenneweg, Kasam & McMullen 2021, chapter 9). A DateTime function used (e.g., Wednesday's entry) in dealing with calculation of days was the addDays() function. This was implemented in the Microflow of the page where it generally made the process of calculating the days from StartDate to EndDate automatically. Without the use of manual coding but rather low-coding, and through the use of these expressions, the automation of these functions was made possible as observed from the daily monitoring.

Automated processes streamlined application building by eliminating the need to create repetitive, basic components that are commonly used in applications. Without the need for

extensive coding, the automation process of the application through the use of Microflow reduces the manual effort and increases the accuracy of work. The drag-and-drop capability and ease of use of these activities makes the process faster in adhering to the requirements of the application (Verveg 3 January 2023).

Part of the process automation of these Microflows, the trainee has come across the extraction of so called sub-microflows while learning through these relevant sources. Sub-microflow represents distinct components or smaller functional segments of a Microflow. Each sub-microflow typically focuses on executing a specific task or function within the broader context of the main microflow, enabling modularity and easier maintenance of the overall process (Mendix Docs 2024d). Through the use of these sub-microflows, this entails improved management of the automation process and is easier to check, maintain and troubleshoot the Microflow itself. This will be a part of the trainee's continuous search for knowledge. Being able to understand the logic behind the sub-microflow, makes it even useful to reuse them in other Microflows. Reading through the Mendix Docs, Microflows should ideally contain no more than 25 elements, such as loops, actions, or decisions. If this limit is exceeded, split the microflow into logical, functional parts by creating sub-microflows, like separating presentation logic from business logic. Exceptions may apply for specific cases like validation or data copying flows (2024e).

As experienced by the trainee while working with Microflow, the ability to create logic for example in validating fields, were deemed useful and efficient in learning how the application would flow. The sources that were recognized in this weekly analysis strengthened the skills and knowledge of the trainee that were demonstrated on these daily monitoring. The observations made were reinforced through the sources and even gave additional information for the trainee to use for the continuation of this diary thesis.

As a reflection, the trainee was able to demonstrate properly the use of Microflow in addressing the requirements asked. It still may need some practice in applying this newly gained knowledge especially on the usage of these sub-microflow in the future, but nonetheless the general grasp and core knowledge were established.

The knowledge gained from this weekly analysis were crucial in building the confidence of the trainee in using them in the continuation of its work and tasks. The learning from the daily entries were utilized and improved together with the knowledge that were adopted through scanning on the sources. It gave a new perspective for the trainee on how to approach its work more effectively in a way that it would apply the automation process principles to be more efficient.

Overall, the daily entries that was exclusively tackled in this weekly analysis reflected improvement on the trainee's part. It had optimized its work and gained insights as to how the Microflow has to be approached and utilized in promoting readability and ease of its use. The knowledge gained from reading through these sources about these activities and functions further boosted the competency of the trainee. It now offers not just a drag-and-drop approach, but a 'drag-and-learn' experience. It will elevate the trainees progress not only in professional development but also in self-improvement.

3.4 Observation week 4

Monday 11 November 2024

Today's work was related to the developmental theme, Mendix Platform, specifically understanding the fundamentals of Mendix and continuous learning on the use of the Mendix Studio Pro.

The trainee's task was to integrate a REST service, particularly JSON into the Mendix application. This involved knowing how and what the JSON structure would be. In this case, the structure was repurposed from a mock application used for testing. The data structure was laid out using the Import Mapping from Mendix. Moreover, finding the Key Id were done as to avoid duplication of objects.

Reflecting on today's activity, the trainee was able to implement a REST service into the application. This experience will be valuable for future tasks requiring similar structures. Through this activity, the trainee developed skills in working with data structure and gained the necessary competency to apply them effectively.

Tuesday 12 November 2024

The developmental theme for today focused on Mendix Platform. It covered the developmental goals which was related to Domain Model and configuring its entities.

Today's task begun with the configuration of entities that were automatically created from the data through mapping in continuation of yesterday's work. The trainee made these entities persistent which means storing them into the database. Since these entities were non-persistent to begin with, recognizing them was part of the trainee's learning process as to how and why we make them persistent. Entities were then created to reflect from the requirements of the application. Proper naming of entities and its attributes were observed by the trainee.

Today's work reflected the trainee's competence in working with entities and its attributes. Errors were avoided because of its prior knowledge in applying the best practices in dealing with them, particularly naming conventions. Learning about persistent and non-persistent entities will strengthen the core knowledge of the trainee when working in the Domain Model of the application.

Wednesday 13 November 2024

Today's work was related to understanding how the layout of pages work in Mendix. This was connected to the proper use of the Mendix Studio Pro as part of the developmental goal of the trainee.

The tasks tackled today was creating pages according to the entities that were created these past days. The trainee was able to implement the concept of low-code with the use of automatically generating the pages from the entities. Configuration of one of the pages were done using the data view and layout grid widgets. The trainee has encountered challenges and errors while using these widgets. It has come to a solution that the layout grid has to be inside the data view in order to gain access to its input widgets. The trainee implemented the visibility setting for the data view based on the value whether the user has an account or not.

Reflecting on today's tasks, the trainee was able to find solution to address the errors that came up. Trial and error were done in the beginning with then combined effort and searching through the forums for acceptable answers. The trainee has gained better understanding while using these widgets, particularly data view and layout grid.

Thursday 14 November 2024

Today's work was related setting and Home page and implementation and configuration of Microflows. This includes validation purposes that will check the validity of data through the use of Microflows.

The trainee's work began with configuration of the home page using a building block called Card Action. The Card Action is a widget as a whole that consists of multiple widgets. These building blocks were then configured and linked to the entities on the Domain Model. They were also linked to the pages that were generated from yesterday's task. Save and Cancel button were created and a Microflow were created along to check for the validation of the data on the pages. Decision and Merge activities were done to check the data along with the validation feedback if ever a user failed to comply with the requirements.

Reflecting on today's work, the trainee was able to use building blocks effectively which obviously made the work faster and easier rather than building them one by one and then piecing them altogether. Working with the Microflow, the trainee was able to apply the skills and knowledge from this week and including the previous week in using and choosing the right activities in implementing the task at hand.

Friday 15 November 2024

Today's work concentrated on configurations of a Microflow which checks whether or not a user has an account. This was a continuation of yesterday's work which was related to the developmental theme concerning Mendix Platform.

The trainee's task focused on building and configuring a Microflow to check if an account existed or if an account is to be created. The Microflow further checked the validity of the account whether they are empty or not. These were managed by Decision activities in checking the validity of the data. Delete Object activity was also created if ever the user wished to delete its account. Debugging was done to address different multiple errors from this week's tasks.

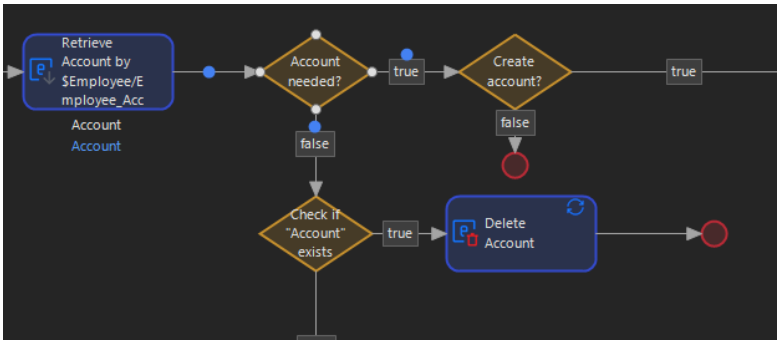

Figure 11. Verifying the existence of an account using a Microflow.

Today, the trainee was able to understand the flow of decision activities to check the validity of the account which was evident in this task. Currently, the experience of the trainee in relation to microflow and debugging errors were time consuming given the fact that sources were limited. Nonetheless, workaround and solutions were found by providing enough time in working with them.

Week 4 Analysis: Mendix Data Validation

The weekly analysis focused on how Mendix approach and apply the validation process to check its data. This analysis covered the developmental theme, Mendix Platform, set by the trainee which focuses mainly on the data validation that were applied on the tasks not only this week but also from last week. This was a continuation of progress in understanding how data validation works in the Mendix environment through the use of their IDE, Mendix Studio Pro. The analysis aims to enhance what was learned by the trainee from days and weeks of tackling with the subject matter. With the weekly analysis, it focused on enhancing the knowledge regarding data validation and understanding the best practices of Mendix for the improvement and skill growth of the trainee.

Through daily monitoring of the tasks, the trainee was able to gather necessary knowledge and skills to better understand the workflow and its process that proved to be helpful for the future tasks. This was reflected from the trainee's performance when working with data validations. Knowing where and when to apply these data validations had such big effect on the totality and outcome of the tasks. Throughout the week, including last week, the trainee had worked with these validations that were applied on a Domain Model, pages and Microflows. The weekly analysis further dived into these matters and provided insights on how to use them efficiently.

Data validation is an essential data quality management process that validates and verifies whether system data meets predefined quality and integrity standards. This ensured collected data is verified and accurate before being used. It typically involved series of documented tests against criteria such as code lists or data types. This process enhances data readability and accuracy by preventing invalid entries. Through this, it minimizes the need for manual checks, streamlining workflows with minimal human intervention. In return, it enhances security through the use of these validation requirements by preventing malicious data entry (TIBCO 2024). With this understanding, data validation is an integral part of any process especially in terms of security. It is important for maintaining data integrity and efficiency that could potentially prevent errors and inconsistencies.

Data validation in Mendix can be applied at the entity level, on pages, or in microflows, following organizational standards. Validation rules ensure data meets specific criteria, such as uniqueness, format, length, or matching predefined value (Kenneweg, Kasam & McMullen 2021, chapter 6). The data validation applied by the trainee were done accordingly and properly. Understanding the rules such as unique, range, maximum length was needed in order to adhere to the specific requirements. Validation rules are conditions that must be met before an object can be committed. If a condition is not met, the server triggers a

validation error. The rules are applied in a specific order wherein if multiple rules are violated, their error messages are recorder and displayed in the same sequence (Mendix Docs 2024f).

Data validation on Entity level were achieved by the trainee through the Domain Model. Validation rules are automatically triggered whenever changes were made to these objects once committed (Mendix Docs 2024g). Working with the entity proved to be challenging since these validations directly apply to these entities. It doesn't seem possible though to have this validation to be reusable to other parts of the project since it is entity based only.

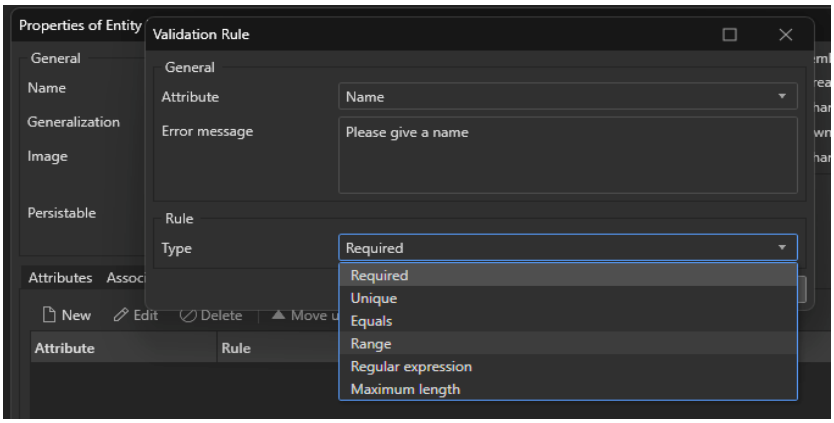

Figure 12. Validation on an Entity level.

Entity level data validations were applied by the trainee during the past weeks working with them but has not applied them on the advanced data validation. Mendix is able to handle complex validations wherein it also happens in the Domain Model but with the configuration of the entity's Event Handlers (Mendix Docs 2024g). Event Handlers provides a flow when handling the validation. It has the capability to check validations before or after data has been committed. The trainee had not applied any event handlers but with the crucial information from these sources, it gave a background on the possibility to apply them in a more complex situation. Event handlers are used to trigger specific logic during database events, such as creating, committing, deleting, or rolling back an object. A microflow is executed during these events and for the 'before commit' handler, it returns a Boolean value. Returning False raises an error to prevent the event, while returning True allows the event to proceed (Kenneweg, Kasam & McMullen 2021, chapter 6).

Validation on pages have two settings namely Type and Message. The type has either no validation chosen which is the default option, a predefined validation or a custom validation. The message is the validation feedback wherein the error message is shown to the end users (Mendix Docs 2024h). Page validations has been applied by the trainee on the input elements of the page. Each input element has its own validation rules as a standalone rule

which displays their own validation feedback or message to check its validity. Though the page validation has not been often used by the trainee, it still good to know that it does exist and has its own purpose that differs from the other validation types.

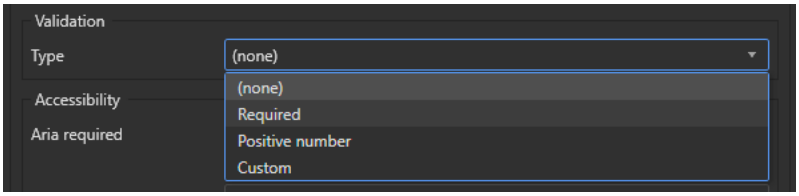

Figure 13. Validation on a Page level.

Microflow data validation is achieved through a trigger, for these weekly entries by the trainee - a button, which was used to call the microflow to check its validity. The trainee has used this method more often than the two validation types since using a Microflow to do validation made the rules more accessible and easier to update and maintain.

Part of the Microflow data validation was evident in these weekly entries including last week's entries. The Microflow data validation was applied with the use of Decision activities. As already known that the decision activities have either a true or false flow, it also has the ability to define its decision type known as 'Rule'.

A Rule is a reusable logic component that can be applied throughout the application, functioning similarly to a microflow. It provides an efficient way to handle validations, reducing the risk of human error or forgetting checks, such as verifying for empty values or empty strings (Kenneweg, Kasam & McMullen 2021, chapter 10). Using a Decision Rule allowed the trainee to write a logic similar to a microflow which were then reused many times in a Decision. With these insights, the trainee was able to understand the use of decision rules as to why is it used. It has limitations compared to a normal microflow in terms of availability of activities to be used but nonetheless, it is aligned with the concept of low-code in component reusability and maintainability. Using this type of rule still demands a lot of effort and time from the trainee. The knowledge and skills acquired from this provided core understanding on how the trainee would effectively apply these concepts in dealing with them in the future.

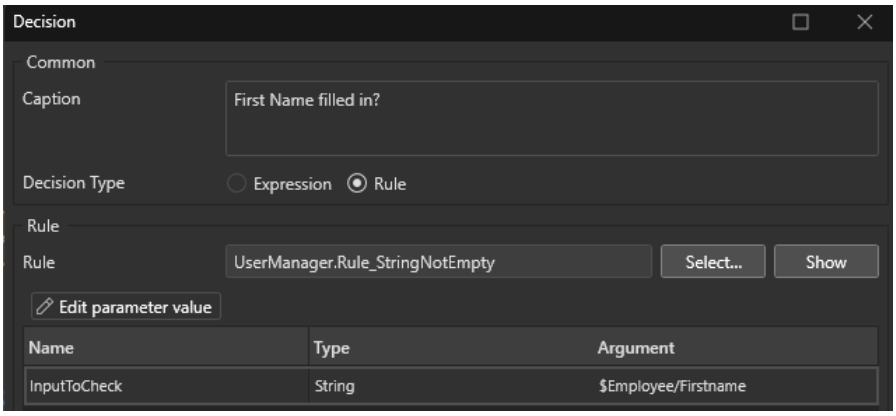

Figure 14. Decision activity which uses a Rule as its Decision type.

Validation Feedback activity performs a validation check and displays a red error message to the user if the validation fails. This message appears below the widget or input field linked to the attribute or association that failed. The validation feedback is configured on the two properties of the activity itself (Mendix Docs 2024f). Data validation feedback was never left out on every validation field. This is crucial part in checking the validity of data since it provides short and meaningful information as to what triggered the error. The feedback relies on the builder of the activity; therefore, it should be clear, concise and helpful in a way that it provides ample direction and solution for the user.

Reflecting on the daily monitoring, the tasks done were proven to benefit the trainee in the long run. The insights gained from the weekly analysis enhances the trainee's competence in dealing with data validation in terms of its application whether be it on entities, pages, or in microflows. These demonstrations of skills were reflected in the daily diary monitoring which reflected the information gathered through the weekly analysis. Challenges faced by the trainee were that of limited source materials but nonetheless, the source materials found served to deliver the knowledge the trainee is seeking. Effective and correct approach to the errors and difficulties faced during the daily monitoring reinforced the trainee's capabilities in dealing with them whenever similar problems are encountered. Through the weekly analysis, the understanding of the trainee about validation of data was enhanced and was an important part in the development of its professional advancements.

3.5 Observation week 5

Monday 18 November 2024

Today's work was directed to Mendix Platform as a professional developmental theme, with the continuous integration of skills with Mendix Studio Pro covering the security settings of the application.

The trainee's task focused on configuration of the default settings set by Mendix Studio Pro. It also covered learning and understanding of the User and Module Roles as to who could have access right to the different parts of the application. The task began with setting the App Security from Off into Prototype/demo security level. The trainee created module roles according to the users of the application in order to set as who should have access on particular parts of the application. Errors have been handled which came from this type of security setting through configuring access right to the pages and microflows that needed to have at least one module role in order to be used or accessed.

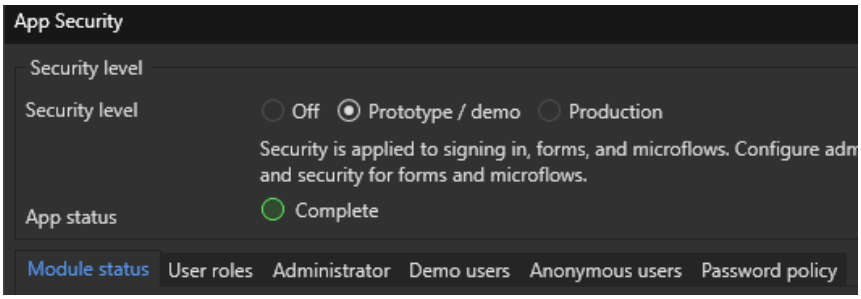

Figure 15. App Security setting set to Prototype/demo.

From today's task, the trainee was able to identify and create different module roles for the application. Further editing and configurations of these roles could happen depending on the application requirements. Errors have been addressed though further understanding as to how these user and module roles work still demands time and practice. The learning from today's goals have provided information about how roles were implemented by Mendix and understanding the security settings that comes with these certain roles.

Tuesday 19 November 2024

The work for today was a continuation of yesterday's work. It was still about the configuration of settings, but this one is specific to User roles. Today's activity still aims to know more about the Mendix Platform as its developmental theme and security settings as its developmental goal.

Today's task focused on creating user roles that were also in the same as App Security setting shown in Figure 15. The trainee's task began with identifying the potential users of the application and created user roles specifically for them. Configurations were done as to what

module roles the user roles can access. Home pages were then created through copy-paste that served as separate landing pages which catered to the specific users. This was configured under the Demo users shown on Figure 15 as well. Production security level was also chosen to further check for errors regarding security access on the pages and entities.

Based on the task by the trainee, it was evident that the trainee was able to demonstrate its knowledge in configuring the security setting and identifying the right users for the application. The trainee was also able to apply the low-code approach of Mendix through re-using components in building user-specific home pages. Further exploration and skills are required though in dealing with Production setting as this required more time to understand.

Wednesday 20 November 2024

Work for this day was still security settings of the application. This was still a continuation of yesterday's work as part of the developmental growth of the trainee in understanding Mendix through its Mendix Studio Pro IDE.

Today the trainee's task was to give page access to the module roles identified. Knowing which pages the users have access to and which pages are restricted were part of the security settings of the application. Microflow access were also configured based on the module roles. This was a bit tricky for the trainee since not all users should have access to the pages and microflows as it could compromise data leakage.

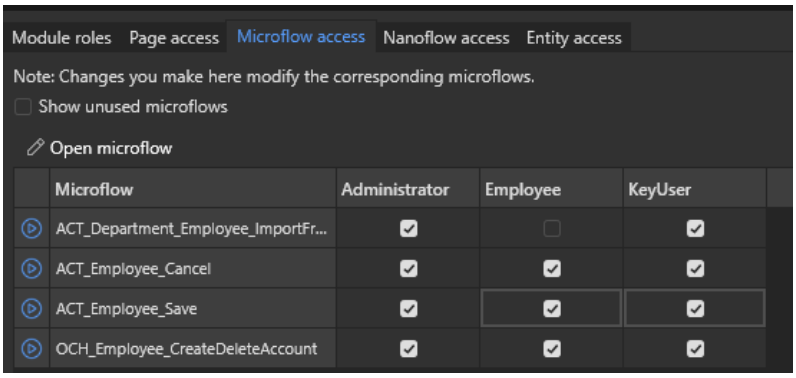

Figure 16. Security settings on Microflow access.

Reflection on the task, it was helpful for the trainee in understanding how setting the security access on pages and microflow were that important. The trainee was able to identify and gave the right access to the users from looking at the errors pane on the IDE. It was reliable for now in providing which caused the error and helped it solve the problem.

Thursday 21 November 2024

The task for today was connected to yesterday's work. It was giving access to users in order to have the right to use the entities. Since the pages, microflows, and entities were all connected, configuring them with the right users are part of the trainee's task for today.

The trainee's task was to give access to the users in the entity of the application. This includes the configuration of roles and their rights in creating and deleting data from the application. Further configuration was done on whether certain users have the ability to read and write on these data. Understanding who should have access and certain rights of these settings was part of the trainee's developmental growth and further understanding of the security measures Mendix recommends. User roles were then updated in regard to which modules roles they could have access to.

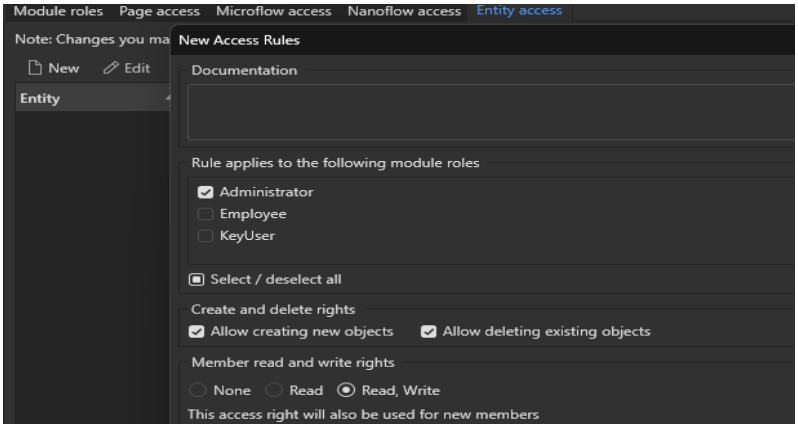

Figure 17. Security setting on Entity access.

With the learning today, the trainee was able to configure successfully the entity security of the application. With the Security level set to Production, there were much to be learned but with the task for today, it gave insight to the trainee how entities could be set properly.

Friday 22 November 2024

Today's work was a continuation of yesterday's work specifically configuration of the Entity access. This was part of the trainee's further exploration about the security settings under entities using an XPath constraint.

The trainee's task began with configuration of the Domain Model to apply a secondary relationship on two entities. This only meant that these entities have another layer of security set for them. The security was set through an XPath constraint which limits the access of certain users for certain data. Access rights were also updated since an XPath has already been

applied. Access rights default setting was updated from Read and Write to Read only and vice versa and also to the Members of the Entity.

Figure 18. Access rights configurations of Members.

Today's work reinforced the trainee's work from these past days of the week and its understanding on the proper application of Security measures according to Mendix. Though it was challenging to know which of these Access rights have to be configured, it was still a success for the trainee to follow the guidelines and implement these security settings.

Week 5 Analysis: Security settings in Low-Code

Daily entries from week 5 emphasized on the Security settings and measures of Mendix concerning its data. The daily diary monitoring covered the subject matter like App and Module security and configurations of Page, Microflow and Entity access. This weekly analysis geared towards the professional developmental theme set by the trainee revolving around Mendix Platform, specifically on the configuration of security settings using the Mendix Studio Pro. The weekly analysis though cantered on these matters, it also focused on further understanding of the trainee in relation of security measures in regard to low-code and Mendix. This analysis served as a stepping stone for the trainee in enhancing its skills and knowledge to properly configure the application and comply with the security best practices enforced by Mendix.

An application security focuses on detecting and resolving vulnerabilities in application software to safeguard against unauthorized access, data breaches, and code tampering. It is a continuous process that plays a vital role in both software engineering and cybersecurity, ensuring the protection of sensitive data and application integrity (Finio & Downie 2024). This understanding of application security was depicted from the daily monitoring of the trainee while working on data which needed security. The way the trainee reflected on this understanding was shown on the configurations of these data in restricting users as to what their limitations were in accessing to them. Though cybersecurity had not been touched by the

trainee, not only this week but also the previous weeks, learning and looking for intel about them would be deemed beneficial for future professional development. Nonetheless, security has many levels and depending on the requirement of the application, it still has to be implemented one way or another.

Low-code platforms often rely in shared service credentials instead of individual user identities creating security vulnerabilities like authorized access, especially during data migrations or when integrating third-party services. The lightweight nature of low-code application, often built on less secure HTTP protocols increases authentication risks. While data customization access can help, limited coding capabilities pose challenges (Groden-Morrison 2024). This was evident from the trainee's work concerning the access rights for each user. The configurations for these data security settings were firstly done under Prototype/demo security level (e.g., Monday's entry) but then access right have to be updated once more when the security level was set into Production. Security vulnerabilities have been evident from the errors that arose which needed to be addressed immediately when set on this security level.

The trainee has had managed different security levels of that are available in Mendix and was able to at least have an idea on the differences of each security level. Mendix offers three level of app security to match a project needs: Off – where no security and is open access; Prototype/Demo – where logins and forms are secured with configuration roles; and Production – which applies full security to all app components, including forms, microflows, and entities, with detailed user role settings (Kenneweg, Kasam & McMullen 2021, chapter 5). Knowing this information, the trainee has had better understanding on which type of security level to use when approaching different applications which requires different types of security levels. The sources gathered further supported the work of the trainee from this week's daily monitoring. By applying the different security levels, Production level required more attention and raised more error to be handled. It was only justifiable for it required protecting customer data and ensured privacy.

In Prototype and Production levels, Mendix allows for demo and anonymous users and configure additional settings like passwords. This helps manage access and enhance security in cloud and local environments (Kenneweg, Kasam & McMullen 2021, chapter 5). Working with anonymous users, the trainee has had only but a minor understanding on them. Knowledge and applying the anonymous user would be a task that would be beneficial for the trainee whenever a requirement of such arises, but for now learning about them would be helpful for future growth. Login credential and passwords were set to default the entire time

the trainee was working. Hence this would be an additional learning that required attention as password is crucial in maintaining security protocols and managing application risks.

Modification on security settings for Application and Module level are connected even though they have different security settings. Application level apply across all modules within the application, while module-level settings are specific to individual modules which entails that a user role could have one or more module roles. (Mendix Docs 2024i). Setting the user roles and module roles were part of the daily entries of the trainee. Creating user roles gave the ability for the application to be used by different users. Module roles on the other hand, was limited to the module that they were created in. The connection for these two roles were configured as to which user role could have access to these modules. This was great in terms of security setting as it gives added security measures more likely double security. This means that if a user role wanted permission to access a module, that user has to be given a module role. They go hand-in-hand as reflected when the trainee set the security level to both Prototype and Production levels which created massive errors as the module required a user role for access capabilities.

In low-code platforms like Mendix, security checks are applied to reusable components, allowing them to be used in multiple applications without re-assessment. However, if security protocols change or the components is updated, it must be re-evaluated. Layered security on the application 24/7 provided top-notched service by Mendix together with its partners. (Baalbergen 30 August 2021). Security settings on reusable components was so helpful as once it was configured, it doesn't need any refinements unless the security level has been changed. With this type of security settings, it caters to different applications depending on how it is managed.

Having a clean environment implies high level functionality of the application. With security measures like removing unused modules, microflows and components, it enhances readability and reduces vulnerability of applications. Configuration of roles and its access right should be strict. (Mendix Docs 2024j). Cleaning the workspace environment was crucial in maintaining security of the application. Unused feature of the application imposes risks and reduces the runtime of the application. Though these would be visible to on the error pane of the Mendix Studio Pro, it is still best practice to avoid unnecessary creation of components if not sure of its use. One thing that the trainee did was to select components then exclude from the project, this way it was retained without creating any warning, but this only imposed risks for the application. As this weekly analysis was learning and searching information from

these relevant sources, adhering and following these best practices will provide better judgment in the trainee's future work and practices.

With this weekly analysis, the learning found, gathered and internalized had strengthened the core perception of the trainee regarding security measures. The trainee has gained valuable insights into implementing security within the Mendix Platform. One of the key realizations has been the importance of security measures in the early application development process to ensure data protection. Throughout the learning journey, the trainee explored various levels of security in Mendix. The configuration of access rights at the modules, entity, and page levels, ensured that permissions were aligned precisely with the requirements. Though further implementation and learning of security measures were required, the learning from this week was supplemental in achieving the level of knowledge the trainee desired. Overall, the trainee's experience has reinforced its understanding of the critical role security plays in application development.

3.6 Observation week 6

Monday 25 November 2024

Today's work was creating a Microflow to go through a list data using a loop activity. This was a continuation of the previous weeks' work. The professional developmental theme of the trainee for this day was in line with the Mendix Platform specifically with the goal of enhancement of skills using Microflow.

The task was focused with going through a list using a loop activity which processes the list one at a time. Depending on the loop type, either the loop becomes an 'Iterator' or also called 'For each' which goes through all the items on the list or 'While' which checks the condition whether true or false. Inside the iterator was placed an activity called Create variable which was set as an Integer/Long with a value of 10. Another Create variable activity was then placed in order to set the Date and Time with a value of [%BeginOfCurrentWeekUTC%] as its system variable. Inside the iterator, a While loop was placed which checks whether the condition was met on these Create variable, meaning to say, the goal was to run the loop 10 times. Then a Change variable activity was placed in order to calculate the Date and Time like StartTime and EndTime. Use of a function on these activities called addHours() was meant to calculate the Date and Time as a reference on the conditions set by these loop activities. When these conditions were met, a Commit activity was then placed last in order to save the data into the system. Domain Model was also updated as part of the loop requirements depending on the behaviours of the object once they were deleted or updated.

Configuring the error message was also done which provided information why a certain data could not be deleted, in this case, it can't be deleted if the data was still associated with another entity object.

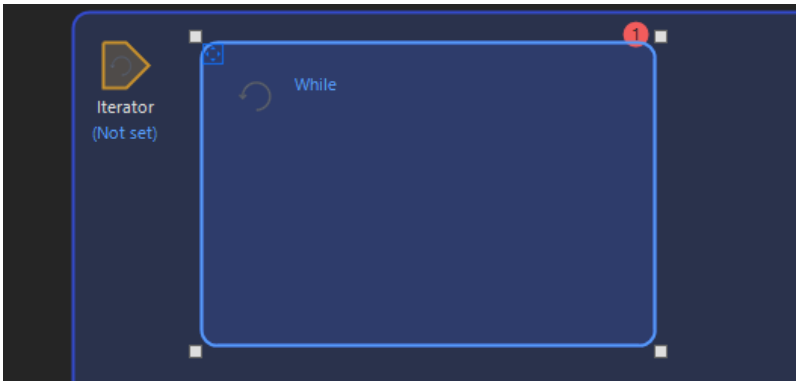

Figure 19. Example of a While loop inside an Iterator/ For Each loop.

The task for today was a helpful for the trainee's competency when working with Microflows. Usage of essential activities like loops provided alternative way how Mendix go through its data.

Tuesday 26 November 2024

Work for today was about an OQL (Object Query Language) counterpart of Mendix for an SQL or Structured Query Language. Part of today's work was for the professional development of the trainee in working with the platform.

The task began with understanding how the syntax works. Installation and importing of the OQL module from Marketplace and its necessary packages like the Community Commons package was also installed. Modules from Marketplace were not always compatible with the version of the layout hence errors would arise from these even if prompt with a successful installation. The error was straightforward, it was addressed by changing to any of the default layout that still exist in the system. The issue with importing and installing modules from Marketplace was that it automatically used the layout that came with the module, so the trainee's solution was to use the layouts that were tagged as default. Queries were then configured in the multiple microflows using activities like Create Object. Challenge faced by the trainee was although there are similarities of SQL and OQL, writing the query was a bit different from each other. One more thing was the trainee's forgetfulness to save its work. When the platform crashed, the trainee has to go back to where it was last saved.

Still, this is a learning phase for the trainee to work with OQL and its syntax and a reminder that a simple step of pressing ‘Save’ could save a lot of time and effort. The lesson gained from this activity today provided inputs for the trainee which points and aspects needed further developments.

Wednesday 27 November 2024

Today’s task was related to the previous week’s entries regarding XPath. The task was not for security purposes but for constraining data shown on pages. The developmental goal for the trainee was to explore Mendix further with the use of an XPath.

The work began with creating a page to show a data. Data Grid was used to show the data requests by the user. Data source was set to use an XPath where either the contents of the data grid will be automatically filled or be built manually. Accepting the prompt to fill the contents automatically will display all the attributes that belonged to the entity. While declining it on the other hand, will display a blank data grid with the possibility to build the data from scratch. For sustainability and ease of use, the trainee automatically filled the data grid then deleted the fields that were irrelevant for the page. Needless to say, that any of the solution it chooses, the displayed data grid will be the same since it was the approach that differs the two. A Tab Container widget was then placed into the page to separate data based on their own pre-defined list of values. Enumeration was then created and used to give the data their own values whether Submitted or Archive. The Data Grid containing the data were then placed inside the Tab Container. XPath constraint ‘Submitted’ was selected to show only the data which were tagged as such.

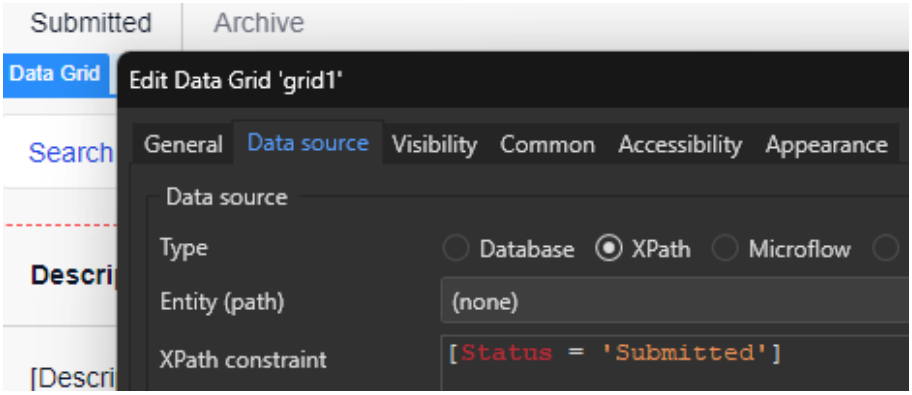

Figure 20. XPath constraint in a Status where Submitted was chosen from the Enumeration.

Reflecting from the task, the trainee was able to use an XPath constraint successfully not only in applying in security but also through the use of an Enumeration.

Thursday 28 November 2024

Task for today concerned about XPath constraint similar to yesterday's activity. The task was configuration of the data shown on the Home page and limiting what belongs to a current user.

Work for today started on setting the association of an entity to the current user. This was done using a microflow with a Retrieve activity specifically in order to retrieve the user's account and associate it to the entity which will then be shown on a page. A button was created and used to call the Microflow. In the Retrieve activity, an XPath constraint was configured. The user was retrieved by using '[%CurrentUser%]', a system variable used in Mendix, as its ID. A Create object activity was then placed in order to set the association of the entity to the current user. The microflow was then configured with a Show Page activity that redirects the action to the page it chose as its value. In this context, what was shown on the page will only be the data that belonged to the current user. This was possible with the constraint that is bound and set only for the user ID.

Using XPath to continuously enhance the trainee's skills skills was relevant to the learning themes and goals that were set. More skills though were required in using these system variables and choosing the right one as there are many and has its own different functions.

Friday 29 November 2024

Today's task focused on the configuration of an entity in connection to its access rights as the User. The tasks further explored the security settings using the XPath constraint for users of this entity. This is in parallel with the trainee's set developmental goals regarding the Mendix Platform which was a continuation of yesterday's task.

The work began by granting the selected User with its access rights whether it could create and delete data objects. This configuration happens under the Entity access. Read and Write access rights were also configured for security purpose. XPath constraint was then used in order to grant access for Users to their own information whether they are the Submitter or Approver of requests. This was done by applying the system variable '[%CurrentUser%]' as previously used in yesterday's entry. Through this security setting, only the current user will have a view of the information that has a direct connection with it. The User was then set as the Submitter with the following constraint – [Management_Submitter = ['CurrentUser%]']. This meant that the User was set as the Submitter which was tagged as the current user. On the other hand, another XPath constraint was made where the Administrator was set as the

Approver and tagged as the current user. In both cases, the XPath constraint checks the rights of each user in getting access to these data.

Today's task contributed to the overall competence of the trainee regarding the use of a system variable and proper use of an XPath. It served to provide the trainee with the essential learning to further understand how system variables could be utilized is providing data security.

Week 6 Analysis: Data Interaction with Domain Model

The weekly analysis concentrated on how interaction of data happens in the Domain Model. The analysis further covered the use of XPath in various data interaction like retrieving data, applying constraints, and security access rules in the Domain Model. The weekly analysis centred on the professional development theme of the trainee regarding the Mendix Platform, specifically how security could be applied, proper configurations of Microflow and further understanding of the Domain Model. The analysis aimed not only to elevate the gained competence of the trainee while doing the daily monitoring but also provide additional insights from the relevant sources identified during the week.

With this week's daily diary, it reflected the trainee's discipline to understand and internalize the Mendix Platform despite the challenges faced. This week was a reflection how the trainee implemented the requirements while working with Microflows, Pages and Domain Model. Proper use of the Mendix Studio Pro software was also utilized keeping in mind the learning obtained from the previous weeks.

XPath is a query language used to retrieve data by selecting objects, attributes, or associations. It is commonly used to define constraints for data retrieval using its unique elements, operators, and functions (Mendix Docs 2024k). As seen from the daily monitoring of the trainee, retrieval of data (e.g., Thursday's entry) was applied using the microflow. An XPath operator was used to select and limit the ID to be only of the [%CurrentUser%] which was Equal to, hence id = [%CurrentUser%]. Counterpart of this operator is the Not equal to or '!='. The two are the most used operators since it they are not limited only to numbers unlike the Less than '<' and Greater than '>' operators which applies only to integers. XPath has a lot of operators and functions to be discovered and applied by the trainee. Other operators and functions would be part of the trainee's continuous learning and growth if ever they come up anytime during the diary thesis.

Retrieve is one the most commonly used Microflow activity among others. Microflows provide a visual way to manipulate data without writing SQL, allowing you to retrieve objects or list of objects (Kenneweg, Kasam & McMullen 2021, chapter 8). XPath can be used in a Microflow using a Retrieve activity as mentioned in the above paragraph. This ensured that once a button or an action triggered the Microflow, the XPath constraints that were configured in the Retrieve activity will take effect thus making it effective in adhering to the rules of the XPath. The retrieve activity will define the data that is returned from the database. This also means that if it the Retrieve does not satisfy the applied constraint in the Microflow, this will result to an error or that there will not be any data to be displayed.

Data interaction happens in pages through the use of an XPath constraints that allows access of data from the Domain Model (e.g., Wednesday's entry). This was done through the manipulation of the Data Grid which houses the data from the Domain Model then was shown in a page. Pages and data grids often apply constraints such as XPath filters on specific attributes which can limit access to the full dataset. Caution is advised when designing these views to ensure appropriate data exposure (Kenneweg, Kasam & McMullen 2021, chapter 10). Limiting the data to be shown was observed by the trainee which was only relevant to the particular user. Not using any constraints in the data would potentially display everything in the page, which means that data was not managed accordingly.

Filtering information could happen in pages through the use of a search bar if only looking for a specific data. Filtering the data could also happen using the XPath constraint through the use of operators and could combine multiple constraints through the use of logical operators – And and Or (Mendix Docs 2024l). Using And operator in combining two constraints results in a list of objects that satisfy both constraints during retrieval of data. The Or operator combines two constraints that satisfies either one of the two. Combination of And and Or operators was also possible thought the use of parenthesis. By placing a parenthesis on one of the constraints means prioritizing it over the other. The information obtained from these sources were helpful and crucial for the trainee in understanding that XPath constraints could behave like this. Using the XPath operators provided alternative ways to approach how data interacts between pages and Domain Model in filtering the information to be displayed.

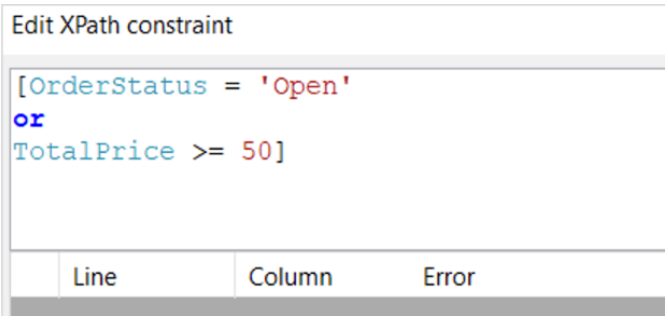

Figure 21. Multiple XPath constraints combined using an OR operator.

Entity access rules define user permission for an entity's objects, including creating, deleting, viewing, or editing attributes and associations. XPath constraints can limit these rules to specific objects; if no constraint is set, the rules apply to all objects (Mendix Docs 2024m). Setting whether a user could be granted with create and delete rights only means that data would not be susceptible to any accidental misuse of the database. The read and write rules for entities defines which of the data could be manipulated by the user. Reading through Mendix Docs, Member's read and write rights control whether users can view or edit an entity's attributes or association. The value '-' means users could not view or edit and is also the default setting. 'Read' means users could only view the data bot not allowed to edit them while 'Read, Write' gives full access to users to view and edit data (2024m). Good thing with Mendix granting Read and Write right to the members could be done with an easier way using the 'Set all to' option then after that, configuration could further happen depending on the access rights of each user.

Entity access was set for security measures using access right and XPath. This was evident with the trainee's work (e.g., Friday's entry) where specific roles were given different access rights and XPath constraints whether the role was set to be a Submitter or Approver. This ensured that only the authorized role, either User or Administrator, can have the power to access the task or view and modify the data to whichever was related to them. The 'Path to user' button creates a link from the XPath constraint to the current object to the current user via an associated entity. This ensures access rules only apply when the user matches the associated user role (Mendix Docs 2024m). The Path to user button is an assistance specific to Mendix, this assists in building the XPath constraint easier and faster in associating the current user to the data object. What the trainee did for the daily entries was the manual way of doing it. It was a way to understand how it worked in the first place then once the logic was established, an easier approach could be done while aiming to accomplish the same result.

To create good and reusable Domain Model, organizing related entities and associations into a clean and manageable structure, using clear, descriptive names for entities, attributes and association enhances readability and maintainability. Utilizing inheritance and generalization also helps reduce redundancy and improve reusability as well as proper implementation of access rules based on the security requirements (Ghartey-Tagoe 21 March 2023). These were somehow observed by the trainee as part of the best practices in naming and creating entities in the Domain Model. Clear and simple yet informative naming convention proved to be useful especially when the application goes to a more complex requirement.

The learning from this weekly analysis strengthened the trainee's core skills and understanding on the use of XPath, not only in security but also in interacting with data from the Domain Model. Going through the daily entries and looking for relevant sources for the weekly analysis was a bit of a challenge because not many sources were available in the topic of XPath. However, learning from the found sources was deemed important and helpful for the growth of the trainee whichever big or small these sources were.

3.7 Observation week 7

Monday 2 December 2024

Today's work centred around Mendix platform, particularly related to styling the application using the Mendix styling editor.

The trainee's task began with the proper setting of the browser to enable seamless local preview of the application. Specifically, adjustments were made to Google Chrome's settings through its developer tools. Disabling the cache within Chrome search engine made it possible to observe the changes in real time without the need to refresh the browser or relaunch the application. This significantly optimized the development process by reducing repetitive actions while improving efficiency while working. In addition to browser configuration, the trainee created a dedicated SCSS text document to be used for a specific module. This file was initially set up locally within the application directory and then subsequently synchronized with Mendix Studio Pro to ensure compatibility within the Mendix environment. The newly created file was then imported to the main.scss file in order to be integrated into the application's styling settings. To enable customization and maintain design consistency, the trainee further modified the SCSS file. This involved defining variables, including colour variables and adjusting the font sizes. These modifications not only improved the visual appearance of the application but also provided a flexible framework for future styling adjustments.

Reflecting on today's task, the trainee's knowledge regarding styling and user interface was improved. The information and idea found from today's work was important for future work in terms of designing and manipulating the user interface of the application.

Tuesday 3 December 2024

Today's work was a continuation of yesterday's activities which concentrated on the styling of the application.

Today's task focused on effectively using the Inspect Page setting of the Chrome browser in navigating specific contents of the page. Determining the right file and configuration of its properties in changing the background colour were made. Once the right colours were made, the changes were then applied and added to the dedicated file that was created from yesterday's task. Modification on different parts of the page were done such as changing styles of the header and buttons. Changes to the header style were made like font-family, font-weight and colour while buttons styles were also updated with its colour, background-colour and how it looked like once hovered.

```
.btn {
    font-weight: 700;
    text-transform: uppercase;
}

.btn-primary {
    &:hover {
        color: $brand-primary;
        background-color: rgb(104, 219, 104);
    }
}

.btn-warning {
    &:hover {
        color: $brand-warning;
        background-color: rgb(206, 122, 122);
    }
}
```

Figure 22. CSS changes made to a button.

Reflecting on today's work, this further improved the knowledge of the trainee when working with user interface. The knowledge gained proved to be useful on how to utilize the Inspect capability of the page then making the changes to the application itself.

Wednesday 4 December 2024

Working with UI Resource Package was the focus of today's work. Creating a separate module and package for customization.

The task today involved setting up a resource package based on the predefined one, to be modified later for implementing the application's style. A module was designated as a

resource model using the ‘Mark as UI resource module’ setting. This allowed the module to function like a standard module having its own designated contents same to a new module but has a Styling setting. Customization was performed using Notepad by duplicating the default contents of custom-variables.scss into the new custom-variables.scss file within the newly created module. The default SCSS contents were then replaced with the following line - ‘@import “../../themesource/designsystem/web/custom-variables.scss”. This provided access to the contents of the new file. The new custom-variables.scss was then configured and manipulated according to the requirements. Additionally, customization was applied to Mendix’s default title settings. A typography.scss file was also created in a similar manner as the custom-variables.scss to configure headers and font styles. The trainee also created a Main Layout to serve as a basis for other layouts to be created. Containers were added to house static images and a Menu bar. Similarly, a layout.scss file was also created to customize the layout’s styling.

The learning from today’s work was crucial for the trainee in understanding how to properly create a module for customization purposes. Creating separate files was more challenging than expected since settings has to be changed every after a new file is created.

Thursday 5 December 2024

Today’s work focused on creating a Building Block to function as a widget in pages. Understanding the fundamentals of Mendix and utilizing the Mendix Studio Pro were the developmental themes for this activity.

The task started with the creation of a dedicated folder for building blocks, which are predefined combination of widgets designed for reuse and efficiency. A new building block template was created and customized by incorporating widgets such as containers, static images, and layout grids. The container, which houses the static image and layout grids, was customized to enhance visual presentation. Specifically, the Spacing property was adjusted to ‘Medium’, and the Shadow property was set to ‘Small’. For the Dynamic Image widget, modifications were made to optimize its appearance. The Display setting was configured as ‘Thumbnail’, and the Width was switched to percentages instead of pixels. This approach allowed the image’s size to be dynamically adjusted based on the context of its placement, while still enabling further customization when applied to a page. The layout grid was then subdivided into multiple portions, creating distinct columns to accommodate texts and buttons. These rows and columns were structured to create a clear layout. A page template was then constructed using the Main Layout that was created from yesterday’s task as a foundation. The page was further enhanced by adding Dynamic Images, texts elements and

buttons. The newly created building block was then integrated into the page through a Template Grid, allowing the contents of the layout with predefined content structures. A template.scss file was created to configure and fine-tune the style settings for the page template. This file followed the same creation methodology as the other styling resources and ensured that the page template was consistent with the design principles across the application.

Reflecting on today's activities, gaining a deeper understanding on how building blocks work proved to be useful in facilitating reusable components. The application of what was done yesterday and today made the page template easier and faster to create rather than creating it from scratch.

Friday 6 December 2024

Holiday.

Week 7 Analysis: UX/UI Design Framework

This week's analysis concentrated on how Mendix integrates its Design framework into the application development process. The analysis enhanced the trainee's knowledge and skillset in dealing with the concept, with the emphasis on user experience (UX) and user interface (UI) design principles associated with styling the application. The analysis formed part of the trainee's professional development theme, Mendix Platform, aimed to explore Mendix's capabilities as a low-code platform while applying this knowledge within Mendix Studio Pro to develop more effective applications as future references.

User Experience or UX Design helps users navigate digital products or processes easily and effectively, delivering maximum value. It is a dynamic field that evolves with changing user needs and solutions, impacting industries like software, web design, and game development. Accessibility is a significant focus, ensuring digital experience cater to diverse needs, such as quick page loading, support for disabilities, and inclusive design. Companies benefit from good UX design as it reduces customer dissatisfaction, increases revenue, and improves overall experience. UX design enhances both user and employee experiences with its effective use of digital tools (Webflow Teams 2024).

The User Interface or UI is the point of interaction between people and technology. UI designers create intuitive features such as buttons, icons, menus, and gesture-based or voice commands to facilitate seamless communication between users and digital products. By

incorporating interaction design principles, it ensures interfaces are logical, engaging and user-friendly. While interfaces may vary in design such as Graphical UI and Voice-controlled Interfaces, their ultimate goal is to create an enjoyable user experience (Figma 2024).

Atlas UI, developed by Mendix, simplifies UX/UI design just as Studio Pro streamlines application development. It enables intuitive design using ready-made widgets, page templates, and building blocks. Atlas UI supports a modular approach allowing reuse of layouts and customize components which once updates are made, it is applied across all modules where they are used (Kenneweg, Kasam & McMullen 2021, chapter 7). This was evident from this week's daily monitoring that reflected the trainee's work with utilizing the pre-defined widgets and building blocks in configuring the page template. This made the work easier and faster to build based on the trainee's experience and these building blocks were made to be reusable components. Mendix Atlas UI is a design framework that simplifies and accelerates user experience. Built on the principles of simplicity, harmony, and flexibility, it offers focus on what matters most, maintains consistency across different platforms, and visually appealing in providing user experience (Mendix Docs 2024n).

The initial step in creating a Mendix app is selecting a navigation layout. This layout acts as the framework and backbone for dynamic pages, either Web or Native, which acts as the structure of the application. The layout is designed using the platform's layout editor, where components like layout grids, placeholders, containers, menu bars, and snippets are used to define reusable page patterns for the application's user interface foundation (Mendix 2024). This was reflected from the trainee's work (e.g., Thursday's entry) where the layout has to be created in order for the page components such as the Dynamic Image and building block to be configured. The layout served as the Master Layout since there were no other layout created. This layout holds only a placeholder from the time it was created until it was populated with different page components. The layout was based on a Web property which was the default setting. The Web layout was typically designed for pages intended to be viewed in a browser or web applications while Native layouts are for pages intended for mobile applications (Mendix Docs 2024o).

Mendix uses pages for building an application's user interface. These pages are created in Studio Pro using layouts and widgets. The main area of the page is editable, starting with a blank template that includes a layout grid featuring one row and a single column spanning 12 columns in width. Built on the 12-column layout principle, the layout grid serves as the foundation for placing all elements on the page (Kenneweg, Kasam & McMullen 2021, chapter 7). While building the pages, the trainee has followed this principle. The 12 columns were the

automatic width for the columns whenever there is only 1 native page. Setting different page components were done through configuring the page component either to Auto-fill, Auto-fit content or Manual. These configurations were applied to different parts of the page upon building of the application. The application for the two settings were more of a trial basis but not for Manual since it was straightforwardly defined. Auto-fill and auto-fit content were most likely interchanged by the trainee based on how the application looked like visually. The Auto-fill option allocated the available space to columns, distributing it evenly while the Auto-fit content options adjust the column size automatically to fit its content. Meanwhile, the Manual option enables the columns to be set manually from 1-12 columns (Mendix Docs 2024p).

Studio Pro and Mendix App Store or Marketplace offers a wide variety of widgets, ranging from commonly used to more specialized ones. These widgets allow the display of data and create interactive elements for end users. In Studio Pro's page editor, widgets can be added in several ways. The toolbar which provides access to the most commonly used data connectors, the Add widget menu which displays all available widgets and Add Building block which creates a building block that can be used on a page (Kenneweg, Kasam & McMullen 2021, chapter 7). The use of widgets has been applied on the trainee's work for a couple of weeks already from the start of the dairy monitoring. Widgets like buttons, menu bars, input, and data widgets have been applied to different tasks and on different page contents. The use of building blocks has been utilized (e.g., Thursday's entry) to speed up the development of pages to be customized. As these building blocks are composed of multiple widgets and different styling, they can be used in a drag-and-drop manner to implement a common UI pattern. As there are many widgets available, the trainee has not applied most of them like the drop-down menus. This only means that there is space for improvement for the trainee in implementing and understanding how the widgets work and how to properly use them. Working with widgets, the trainee has mostly used the Toolbox in looking for widgets. The information from these sources reinforced how the trainee would approach them in an alternative manner.

Mendix leverages Flexbox to ensure widgets remain responsive. Flexbox enables a component to define the layout of its child components, maintaining a consistent design across various devices. For native mobile apps, native layouts simplify the integration of navigation and settings optimized for native features (Mendix Docs 2024q). Though the trainee has not worked with native mobile applications during the diary thesis, this information will be beneficial. The trainee has had prior knowledge about Flexbox but customization about it in Mendix has not been applied. Revisiting this type of information and further understanding of it would be part of the trainee's developmental growth.

Widgets can also be customized using the style objects, which are JavaScript objects containing attributes specific to each widget, often incorporating Native properties like ViewStyle, TextStyle, ImageStyle, and Colors. ViewStyle allows adjustments to general visual elements such as boarders, opacity, layout, shadow, and transformation. TextStyle enables styling of text elements by modifying font properties, selection behaviour, and layout. ImageStyle provides control over image properties, including size, borders, and layout. Colors allow customization of color using RBG notation, hue adjustments, and other modifications. These tools offer flexible precise options for tailoring the applications design (Mendix Docs 2024q). The customization of these properties was reflected from the trainee's work during the week. It was done with creating separate files such as typography, template, and custom-variables in order to be modified according to requirements. The way these separate files were created, it was much easier to detect which page components belong to which file, hence, configurations happen at a much faster pace and with less potential errors.

Best Practices in Mendix has been adopted from guides that were currently being used by Mendix themselves as a guide for designers. This will also benefit not only the trainee, but also the readers that would find their way here in the diary. Some of the crucial information will be further elaborated below.

Avoid cramming all menus, features, and buttons onto a single screen, as it leads to poor user experience. While interfaces may start simple, adding features over time can overwhelm the screen, making it cluttered and less user-friendly. Avoid overcrowding drop-down menus by organizing options logically and prioritizing frequently used features for easy access. A well-balanced interface enhances usability, ensuring users can quickly find what they need. Avoid using long lines of text and small font sizes, as they hinder readability. A good rule of thumb for web typography is to keep lines under nine words, making it easier for users to move their eyes to the next line and maintain smooth reading experience. Design forms with care by using a single-column layout for input fields, as it helps users maintain a smooth flow while completing the from. For selecting between 2-5 options, it is better to use radio buttons instead of drop-down menus, as they require less effort and improve user experience. Ensure each button's purpose is clear by using consistent design elements across the app pages to navigate easily. Proper positioning of buttons, colors, and labels determine their respective functions as it has to be clear and definitive (Mendix Docs 2024r). Some of these best practices has been observed by the trainee, specifically when working with buttons and texts. Making a page simple and not overcrowding it was somehow challenging but proved to be important for user experience. Working with customizing colors has also been a challenge

but continuous learning and application of knowledge found on these sources proved to be helpful for the trainee.

Overall, this week's analysis centred on the customization of page components' design, providing essential insights that contributed to the ongoing development process. The trainee gained a deeper understanding of effectively applying UX/UI concepts within Mendix, particularly in areas like widgets, building blocks, and layouts. These skills will serve as a strong foundation for the trainee's future work, enabling more efficient implementation and design practices. Additionally, continued learning about these elements will ensure smoother integration of user-centred design principles into the application.

3.8 Observation week 8

Monday 9 December 2024

Today's work was a continuation of last week's activities, focusing on the UI Resource package. The goal was to implement a version tracker for the application to provide information for future edits and documentation.

The task began with identifying how to indicate a version of the module that was being developed. This was achieved by adding a Constant to the module. The Constant serves as a configuration value that defines and tracks the module's version. Mendix employs semantic versioning, which comprises three components - the Major version, Minor version and the Patch version. Following Mendix's recommended approach to versioning, the trainee was able to follow this approach and versioned the application as 13.1.0. Another method to identify the module version was to include it in the Application Directory, where files from last week's tasks were also stored. Additionally, documentation of the work regarding UI was included in the Constant's documentation input filed. This ensured that the versioning process was well-documented and aligned with Mendix standards.

The trainee was able to implement module versioning in two different ways. The first method was through the Mendix Studio Pro using a Constant, while the second involved specifying the version in the Application Directory. Today's learning experience provided helpful information for the trainee into adhering to guidelines and proper versioning of application modules.

Tuesday 10 December 2024

Today's work focused on Mendix Platform, particularly on importing data and debugging.

Daily efforts centred on importing data from an Excel file using the Excel importer installed from Marketplace. Additionally, the MX Model Reflection module was also installed from marketplace to enable data access alongside the Excel Importer. Warning and Errors arose during the installing of these modules, even though they were added as new modules. To ensure compatibility with the application, updates were required for these modules, including adjustments to widgets and design properties. The errors pane provided detailed information about the issues and suggested potential solutions. Updating the widgets were easier as it allowed batch updates with a single click rather than requiring individual updates. Furthermore, access rights were given to the right user roles for these two imported modules, enabling proper configurations and data synchronization.

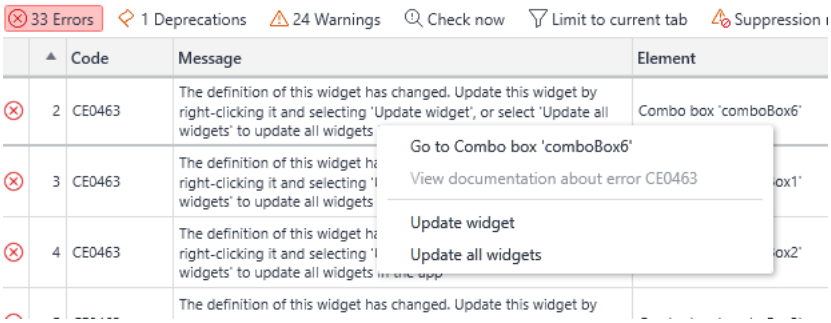

Figure 23. Debugging errors with 'Update all' solution.

The daily activities highlighted how Mendix effectively identifies and resolves errors. This showed how Mendix works as a low-code platform in accelerating debugging process. With the solutions provided by Mendix, the trainee was able to address errors successfully and efficiently.

Wednesday 11 December 2024

The work for today centred on consuming a REST service and creating a Microflow to configure the data to be shown on a page.

Today's task began on working with Import Mapping, a feature first encountered on the previous weeks. A JSON structure was selected as a Schema source, along with its relevant elements. For the REST location, a Constant was used to define its default value, and then a Microflow was configured to enable the REST to call itself. A 'Call REST service' activity was added to the Microflow wherein the location was set to the previously created Constant. The activity's HTTP headers and Response tab were configured with the appropriate username

and password. A ‘Show page’ activity was then added to display the import results to an overview page. Finally, module role setting was also updated to define which roles could have access to the microflow.

Reflecting on today’s activities, the trainee successfully revisited and applied knowledge gained from the previous week to the task at hand. The approach to the matter was quicker, and handling errors became manageable due to similar experience encountered before.

Thursday 12 December 2024

Today’s work focused on REST services, specifically understanding how to expose a REST service and publish a REST service.

The task began by creating a folder named Publish within the Resources folder. The related entity was then selected and connected to the pane of the Publish REST Service to configure its Message definitions. Selecting the appropriate entity and its properties defined the data that are being sent to and received from the app. Key attribute were specified with two operations - ‘Get all’ (to retrieve all records) and ‘Get by key’ to fetch a specific record based on a unique identifier). The corresponding module roles were configured and mapped together with their own designated User roles. These User roles were then granted the necessary permission to have access on the REST Service. The application was set to ‘Run locally’ and the designated Location URL for the REST Service was opened. It was important for the trainee to keep in mind that the link location will not function unless the application is actively running. This solved the error that the trainee faced when trying to expose the REST service and tried the operations mentioned.

With the learning today, the trainee learned how to properly expose and publish a REST service by understanding precise steps, such as defining message definitions and configuring operations, and then aligning module roles to user roles to manage access permissions. This learning experience provided valuable insight into creating functional and secure REST service.

Friday 13 December 2024

Today’s work focused on Exporting Data using an exporter module. This focused on the Mendix Platform as part of the professional development growth of the trainee.

The task began with installing the Excel Exporter module from the Mendix Marketplace, which was necessary to generate document overview using the Excel functionality. Security

and user roles were configured to ensure proper access control to the module. An entity was created in the Domain Model to represent the Excel document, using SystemFile.FileDocument as its source. Additionally, Read and Write access permissions were configured for the document, tailored to different user roles. The trainee proceeded to create an Excel Export Template and carefully selected the input objects to be selected. Selecting the correct document type was a crucial step, as the Excel version had to be 2007 or higher to ensure compatibility. The template was then configured with data structure that matches the intended export format, ensuring that the Excel document would correctly represent the desired information.

The learning experience for today was valuable in enhancing the trainee's competence in navigating the Marketplace and utilizing its available modules, especially the Excel Exporter module. This process reinforced the importance of aligning security, entity configurations, and template design to achieve functional and accurate Excel exports.

Week 8 Analysis: Error Handling and Debugging

This weekly analysis centred around the concept of Error Handling and Debugging. It explored how Mendix approaches and address these topics and examined how these practices are unique to a low-code platform such as Mendix. This analysis further strengthened the trainee's professional development theme around Mendix Platform in fully understanding how errors should be handled properly and how debugging works and delivered at a much faster and easier pace. The analysis does not only reflect this week's entries but also the previous weeks wherein the trainee encountered issues regarding Errors. The weekly analysis aimed to give insight and learning to the trainee in providing a way to learn how to address them effectively and efficiently.

Debugging is the process of detecting and fixing coding errors to ensure a software runs flawlessly and as expected by the developers. It involves identifying issues, analysing the root cause, and implementing fixes using various approaches such as backtracking and automated tools. Integrated development environments and log analysers often assist with debugging. This process works alongside testing in delivering reliable and high-quality output as desired (Quiroz-Vasquez 2024). This understanding of debugging was brought to the trainee's experience while working with Mendix. Though Mendix allows saving an ongoing work even if with errors and issues, running the application platform could not be done unless error handling take place first.

Error Handling in Mendix can be managed through multiple ways. As far as the trainee has experienced, it depends on which part of the application you are working with. Some error handling could be done within the Microflow or could be done on the administrative side of the application. Either way, any of these errors were reflected and shown on the Error pane where every information about them could be read. There are three types of messages that can be seen in the pane. These are Errors which prevents the app from functioning properly, Deprecations highlights deprecated features which may cause issues once newer versions comes in, and Warnings which are not critical but detrimental to user experience (Mendix Docs 2024s). With these types of error messages, only Errors have been tackled by the trainee as it focused more on the functionality of the application rather than warnings that does not have much impact on the application itself. Though with the understanding of these relevant sources, it looked like that the more the trainee ignores these warning and deprecations, it is more likely to lead into bigger issues in the future. This provided a learning curve for the trainee to better address them whenever opportunity arises.

Defensive programming involves anticipating potential issues such as bugs or unwanted data and ensuring that the software behaves predictably or as it was intended to in order to avoid discrepancies in user experience. This approach, which has proven an important role in software development for years, helps manage unexpected errors before it even occurs. Mendix offers built-in features like this such as safely handling empty lists. Implementing defensive strategies and precautions helps maintain functionality and avoid faults that could impact the application (Kenneweg, Kasam & McMullen 2021, chapter 10). The defensive programming was encountered by the trainee while working with Microflows the past weeks wherein it checked whether the input had an empty value or not. This concept of Defensive programming was new to the trainee and the learning from this diary monitoring proved to benefit for the trainee’s future work not only in Mendix but in programming as a whole. The concept of empty and blank (‘ ‘) and knowing when to use them is crucial. An empty check verifies if an object or attribute exists, which means space and database is allocated. A blank (‘ ‘) check ensures that an object’s attributes are not empty, ensuring that values are assigned. Both of these checks prevent issues when data is incomplete or improperly utilized (Kenneweg, Kasam & McMullen 2021, chapter 10).

Working with Microflow in Mendix, error handling options include Rollback (default), Custom with Rollback, and Custom without Rollback. By default, when an error occurs, Mendix rolls back all changes, reverting the process to its previous state and terminating it. This option provides a general error message and demands more to debug. The Custom with Rollback option in Mendix is similar to Rollback but allows the Microflow to continue after rolling back

the changes. This is done in order to perform additional tasks while still undoing the changes made during the process. On the other hand, Custom without Rollback allows changes made before an error occurs to be committed in the database, given the idea that only actions after the error are reverted. It provides flexibility in handling errors without undoing successful operations (Kenneweg, Kasam & McMullen 2021, chapter 10). While working with Microflows, the trainee has had experienced dealing with these errors and was faced with Rollbacks. The way Mendix deals with these errors provided multiple ways to handle the data the right way. In general, Rollback has been the default approach made by the trainee since it didn't dive deeper into dealing with error handling. It was not much given time to learn more about the two different approaches since the Error pane give such informative message as to how an error could be potentially solved. Understanding these different approaches would provide the trainee the needed competence to effectively use them and why it was being used in such cases.

Mendix Debugger is a powerful tool for testing functionality during development. By adding a breakpoint to a microflow, one can pause the operation and execute a specific point and inspect a flow step by step. Once the breakpoint is reached, it halts the microflow and highlights the current activity with a red outline. The microflow's name will also appear in the Debugger window, allowing for detailed debugging and ensuring the process works as expected (Kenneweg, Kasam & McMullen 2021, chapter 10). This concept has been applied mostly since this was the only approach known by the trainee in error handling. With the breakpoint applied, it stops the flow of operation whenever it hits the breakpoint itself. This is done by whenever the trainee thinks that error could be found within this flow. Effective use of breakpoint can be done at any point in the microflow, this is only possible while running the application (Mendix Docs 2024t). This information was a big part in executing the breakpoint itself. At first, it didn't come to the trainee's senses that the application should be running for the breakpoint to take effect. This was an additional error handling for the trainee simultaneously while looking for answers about solving the actual errors.

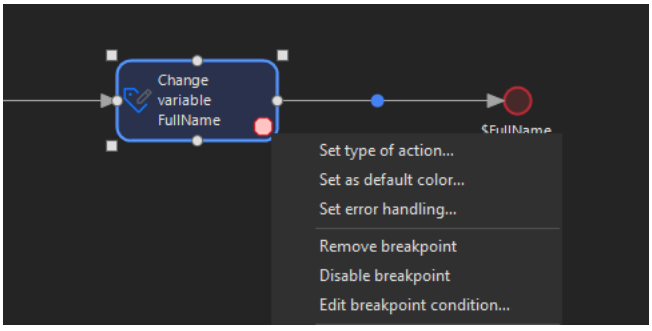

Figure 24. Breakpoint represented by a red circle in a Microflow.

While using the Debugger, proceeding with the flow may come in handy and on what to do next. Possible steps and options are available for navigating the Error pane. ‘Step into’ allows the action to move forward; ‘Step over’ skips checking sub-microflows; ‘Step out’ exits a sub-microflow and resumes the operation to the main Microflow; and ‘Continue’ runs the process until the next breakpoint. Additionally, the Variables window provides real-time contextual information about the user, session, and microflow. These tools overall enhance debugging precision and helps in dealing with errors effectively (Kenneweg, Kasam & McMullen 2021, chapter 10). The steps were familiar to the trainee but utilizing them effectively fell short when using breakpoints. It came to the trainee’s experience that using ‘Step into’ was the most natural way since it follows the flow of sequence in the Microflow. Understanding and going through these sources gave critical information what each of these steps were really meant to do. Nonetheless, the Variables window has been utilized properly by the trainee, following the recommended debugging feature that the window should always be either open or side-by-side with the Error pane in order to follow through with the solution.

As reflected from the daily monitoring this week, the Error pane gave descriptive information about the errors at hand. Mendix as a low-code has been efficient in addressing these with their capability to solve them with just a click as seen on this week’s Tuesday entries. With the possibility to ‘Update all’ saved time and provided acceleration of work and focus to other tasks. This falls under the hood of troubleshooting but the same idea is applied here, solving errors as early as possible as to not stumble around when they are all piled up. From time to time, checking manually your code is essential for quality development. Performing functional testing ensures the code meets user story requirements. Making testing as a habit will empower responsible development practices which in return improved code quality and efficiency over time (Kenneweg, Kasam & McMullen 2021, chapter 10).

Debugging in Mendix can be optimized with best practices and some of them will be tackled. For the After Startup microflow, an application might fail to start. Calling it from a temporary button for testing might do the trick since breakpoints does not trigger due to runtime design. Always check for null objects before using ‘Change Object’ activity by placing a Decision in between to prevent errors. Additionally, for string validations, using a ‘trim’ function combined with checking for an empty object, ensures better error handling (Tripathy 20 June 2022). With an effective error handling, always log errors and stack traces for troubleshooting. Avoiding the use of ‘Continue’ option on the Error pane as it skips error reporting entirely. Incorporate custom error handling, especially for integrations and email-related processes to enhance reliability. Moreover, keep error handling logic simple to avoid slowing down the performance of the platform (Mendix Docs 2024u).

With the learning of this week's analysis, it has equipped the trainee with valuable insights, forming a strong foundation for effectively addressing future challenges especially in error handling and debugging. The analysis provided information that was new to the trainee especially in other ways to approach the issues that came along while working with the tasks. Additionally, the knowledge gained significantly enhanced the trainee's problem-solving skills, enabling a systematic approach to debugging. By leveraging these learning, the trainee has not only refined technical competencies and enhanced its skillset, but also developed a mindset for continuous improvement, which is essential for long-term benefits for the trainee's career.

4 Discussion

The daily entries together with the weekly analysis compiled during the diary thesis monitoring reflected the trainee's substantial growth and development in multiple aspects, both professionally and personally. This offered a pivotal learning experience, imposing great value of firsthand exposure to the Mendix platform and the broader field of Low-code development. Through practical engagement with the different aspects of the application development such as domain model and microflows to name a few, the trainee gained deeper insights into the capabilities of Mendix technology and its effects and approach in business processes. Moreover, the iterative nature of diary writing provided a unique opportunity to track progress, reflect on challenges, and enhance learning, grasping and understanding capabilities of the trainee. These entries highlight not only the technical skills acquired but also the continuous improvements made in order to adapt to changes and have a productive mindset.

4.1 Development Progress and Learning Process

With the time and effort spent working with Mendix platform, the trainee has developed significantly in both technical and soft skills. Initially, the trainee has had limited experience and knowledge while working on Mendix and about low-code in general, but through hands-on practice and diligent learning, the trainee involved himself to an in-depth understanding of how Mendix works including its core features and using this learning to build scalable applications. The trainee enhanced its abilities in designing and developing applications more efficiently by leveraging Mendix's environment in its unique approach in building solutions without deep coding expertise. Moreover, the trainee has developed strong problem-solving and troubleshooting skills, which were essential when navigating through the challenges of building and implementation. This served as a backbone for the trainee on how to move and approach each task daily with more precision and in a less time-consuming manner. The capacity to adapt to new tools and methodologies has also improved more so to become increasingly proficient at learning new concepts and applying them effectively in real-world scenarios.

On a personal level, the trainee developed a progressive understanding on its learning process. Through writing this diary thesis, the trainee has been able to track its own growth, identify areas of improvements, and reflect on how relevant skills have evolved over the course of the diary monitoring. This reflective practice has helped the trainee to become

more self-aware of its strengths and weaknesses and reinforced its ability to cope up and grow continuously in developing its career path.

Learning from the diary-based thesis was more of insightful and interesting than stressful in the end. The trainee has gained solid foundation using the Mendix Studio Pro in its collaboration involving various skills and working in an agile environment. It has fostered critical thinking and reflective learning especially when working with domain models, automating processes using microflows, and building the application itself. As the trainee kept in mind the professional development themes while doing work and diary entries, it has developed an evolving approach in writing the thesis to support its learning. In the beginning, a lot has to be done and changed when writing the thesis, and even on how to write it. But eventually, with proper guidance and keen observation, the trainee was able to find a clear structure and rhythm. Through feedback and consistent reflection, the trainee learned to organize its thoughts, articulate its experiences, and effectively documents growth. What initially felt challenging gradually transformed into a valuable learning experience, helping the trainee to develop confidence in both technical and writing skills.

Searching for relevant and informative sources while working provided ways on how the trainee could improve and develop its thinking and working habits. The diary thesis also helped the trainee to have discipline and concentration to whatever task was on hand. This was observed during weekly analysis wherein the sources either reinforced the learning or provided alternative aspects and concrete concepts. Based on the professional development themes, Scrum methodology was densely focused on week 1. The emphasis on Scrum methodology laid an understanding of its principles and practical application. While the focus shifted slightly in the following weeks, the trainee was able to build on this knowledge progressively. Learning about Scrum artifacts such as Product Backlog, Sprint Backlog, and Increment (week 1 analysis) helped solidify the trainee’s ability to prioritize tasks, plan efficiently and measure progress. With this learned nature, the trainee was able to grasp the importance of agile development and apply it to its work and in the thesis itself, highlighting the value of adaptability and continuous feedback for improvement.

Mendix platform has been the bulk of the diary thesis. It started with learning the basics of Mendix and laid as the foundation for the entire duration of monitoring. Learning how to operate the Mendix Studio Pro from understanding the basic concepts to more advanced functionalities such as integrating third-party apps from the Marketplace (week 6 and week 8), showed how far the trainee has improved. Integrating applications from Marketplace has been challenging for the trainee since it came with a lot of errors. Successful integration

happened only because of previous experiences and searching for solutions from across different sources. Error handling (week 8 analysis) was further discussed on how to deal with different error using various approaches and solutions. As much as debugging was a big part all throughout the diary monitoring, Security was also a huge part in every implementation. Adhering to proper security practices became a priority, particularly in areas like access controls and user authentications. With the right implementation made by the trainee, sensitive data was protected and only accessible to authorized users. Together with security settings, validations on Microflows were applied to prevent malicious data inputs. With these understanding, it has helped the trainee in many ways possible in order to move forward with the tasks. Knowing the best practices and applying them has had an effect on the overall result of the application.

Working with Microflow provided practical experience for the trainee, involving successful use of activities, decisions, loops, XPath and validations (week 3 to week 6). This engagement allowed the trainee to better learn the logic behind automating processes within the Mendix platform. By implementing decision-making workflows, handling iterations through loops, and applying constraints for data filtering, the trainee has structured an approach to solve different application scenarios. Additionally, integrating validations ensured reliability while developing applications. Alongside Microflows, the trainee also developed a solid background of the Domain Model, which serves as the backbone of Mendix applications. By designing entities, attributes, and associations, the trainee learned to structure and organize data effectively according to requirements. Having a wide knowledge about Domain Model impacted the overall outcome, including how the Microflow works. With this knowledge in mind, correct and solid building of the Domain Model came in first as it provided the needed support in creating Microflows through data retrieval, manipulation and validations. This marked a significant growth for the trainee in applying theoretical knowledge into actionable solutions and examples while strengthening problem-solving skills.

Throughout the diary monitoring, UX/UI design played a significant role in ensuring that the applications were both functional and beginner-friendly. Proper utilization of the guidelines and best practices provided by Mendix (week 7 analysis) were properly observed, most but not all. This provided experience for the trainee in creating intuitive interfaces and responsive layouts. Making simple and clear functionalities provided faster and smooth experience for users which increased not only user satisfaction but also accessibility and visual appeal.

4.2 Skill Development and Future Growth

To further develop the skills the trainee has gained during the diary thesis, focusing on continuous learning and practical application of the knowledge would be an initial step. Deepening the expertise on Mendix platform through exploration on the areas wherein the trainee was not able to dive deep enough such as custom widgets, integrations with external systems and increasing optimization of performance. Participation in professional development opportunities and engagement of the trainee to meetings and seminars in regard to updates of Mendix. Continuing learning and obtaining certifications in which the trainee aims to obtain a certificate for Advanced Mendix Developer at the time of writing this diary thesis. Staying updated with industry trends and community forums while engaging with collaborative projects within agile environment to refine the trainee's knowledge and skillset.

Learning about UX/UI has been both interesting and comprehensive, demonstrating its importance while working with Mendix. The knowledge obtained through the daily entries and weekly analysis regarding UX/UI extended beyond Mendix itself. Another thing that proved to be of importance was Security settings, interestingly, through this weekly analysis, the trainee has understood deeper about security measures and on how security should be approached in every situation. These concepts are not only limited to Mendix but can be applied to any development-related projects. The general understanding of UX/UI and Security obtained during this process provided importance and benefits for future career development.

Having the diary thesis with its weekly analysis offered enormous benefits for the trainee to continue its learning. Analyzations of the daily tasks and writing it down made the activity to materialize and gave focus for the trainee where to look and in which aspect to focus. Through writing, it provided the trainee a flow on how to approach the tasks and documenting them gave the trainee a sense of fulfilment. The weekly analysis provided theoretical concepts that complimented the practical tasks, helping the trainee to bridge the gap between the goals and aimed skills. With this approach, the trainee was able to evaluate different methods and make informed decisions. It served as both a tool for self-improvement and means to track growth, offering value to the trainee's professional journey.

Above everything else, the trainee has learned about time management, critical thinking, and ability to adapt to situations and work to its advantage. Being consistent with the strategies learned and having the discipline made the trainee in such a way to be sustainable and versatile in developing its mindset about Mendix and Low-code.

Sources

Akondi, G. 28 August 2020. Part 4 – Application logic using Microflows in Mendix. Xebia Creating Digital Leaders. URL: https://xebia.com/blog/application-logic-using-micro-flows-in-mendix/. Accessed 10 November 2024.

Baalbergen, F. 30 August 2021. Low-Code Security by Design with Mendix's Low-Code Platform. MX Blog. URL: https://www.mendix.com/blog/security-by-design-with-mendixs-low-code-platform/. Accessed: 23 November 2024.

Blanchard, S. 2024. Project Management Basics, module 13. Kirkwood Community College. E-book. Accessed 26 October 2024.

Bloomberg, J. 2020. Low-Code for Dummies, VeriTran Special Edition. New Jersey. URL: https://www.scribd.com/document/650854965/Libro-Low-code-for-Dummies. Accessed: 2 November 2024.

Figma 2024. What is UI design? URL: https://www.figma.com/resource-library/what-is-ui-design/. Accessed: 7 December 2024.

Finio, M. & Downie, A. 5 June 2024. What is application security (AppSec)? URL: https://www.ibm.com/topics/application-security. Accessed: 23 November 2024.

Ghartey-Tagoe, K. 21 March 2023. Improving your Entity Security and XPath Constraints in Mendix. Medium. URL: https://medium.com/@kofighartey-tagoe/mastering-entity-security-and-xpath-constraints-in-mendix-348a8fc71a6e. Accessed: 30 November 2024.

Gosiewski, M. & Swiatek, B. 2024. Mendix platform 101: What is it, why is it interesting, and how much does it cost? Pretius Blog. URL: https://pretius.com/blog/mendix-platform/. Accessed: 2 November 2024.

Groden-Morrison, A. 2024. Top Security Concerns For Low-Code And No-Code Development. Alpha Software Blog. URL: https://www.alphasoftware.com/blog/top-security-concerns-for-low-code-and-no-code-development. Accessed: 23 November 2024.

IBM Could Education 2022. Low-Code vs. No-Code: What's the difference? URL: https://www.ibm.com/think/topics/low-code-vs-no-code. Accessed: 17 October 2024.

Kenneweg, B., Kasam, I. & McMullen, M. 2021. Building Low-Code Application with Mendix. Packt Publishing. E-book. Accessed: 2 November 2024.

Mendix 2024. Experience the Most Extensive Low-Code IDE. URL: https://www.mendix.com/platform/ide/#:~:text=Mendix%20Studio%20Pro%20is%20a%20visual%20model%2Ddriven%20IDE%20with,applications%20from%20a%20single%20model. Accessed: 23 November 2024.

Mendix Docs 2024a. Microflows. URL: https://docs.mendix.com/refguide/microflows/. Accessed: 9 November 2024.

Mendix Docs 2024b. Entities. URL: https://docs.mendix.com/refguide/entities/#:~:text=An%20entity%20usually%20represents%20a,entity%20are%20described%20using%20attributes. Accessed: 23 November 2024

Mendix Docs 2024c. Validation Feedback. URL: https://docs.mendix.com/refguide/validation-feedback/. Accessed: 16 November 2024.

Mendix Docs 2024d. Extracting and Using Sub-Microflows. URL: https://docs.mendix.com/refguide/extracting-and-using-sub-microflows/. Accessed: 10 November 2024.

Mendix Docs 2024e. Mendix Best Practices for Development. URL: https://docs.mendix.com/refguide/dev-best-practices/. Accessed: 8 December 2024.

Mendix Docs 2024f. Validation Rules. URL: https://docs.mendix.com/refguide/validation-rules/#1-introduction. Accessed: 16 November 2024.

Mendix Docs 2024g. Setting Up Data Validation. URL: https://docs.mendix.com/refguide/setting-up-data-validation/. Accessed: 16 November 2024.

Mendix Docs 2024h. Properties Common in the Page Editor. URL: https://docs.mendix.com/refguide/common-widget-properties/#validation. Accessed: 16 November 2024.

Mendix Docs 2024i. Security Model. URL: https://www.mendix.com/evaluation-guide/enterprise-capabilities/security/security-model/. Accessed 23 November 2024.

Mendix Docs 2024j. Implement Best Practices for App Security. URL: https://docs.mendix.com/howto/security/best-practices-security/. Accessed: 23 November 2024.

Mendix Docs 2024k. XPath. URL: https://docs.mendix.com/refguide/xpath/. Accessed: 30 November 2024.

Mendix Docs 2024l. Filtering Data on an Overview Page Using XPath. URL: https://docs.mendix.com/refguide/filtering-data-on-an-overview-page/. Accessed: 30 November 2024.

Mendix Docs 2024m. Access Rules. URL: https://docs.mendix.com/refguide/access-rules/. Accessed: 30 November 2024.

Mendix Docs 2024n. Atlas UI. URL: https://docs.mendix.com/howto8/front-end/atlas-ui/. Accessed: 7 December 2024.

Mendix Docs 2024o. Layout. URL: https://docs.mendix.com/refguide/layout/#master-layout. Accessed: 7 December 2024.

Mendix Docs 2024p. Layout Grids. URL: https://docs.mendix.com/refguide/layout-grid/. Accessed: 7 December 2024.

Mendix Docs 2024q. Widget Styling Guide. URL: https://docs.mendix.com/refguide/mobile/designing-mobile-user-interfaces/widget-styling-guide/#layout-grid. Accessed: 7 December 2024.

Mendix Docs 2024r. Implementing Best Practices for UX Design. URL: https://docs.mendix.com/howto/front-end/ux-best-practices/. Accessed: 7 December 2024.

Mendix Docs 2024s. Errors Pane. URL: https://docs.mendix.com/refguide/errors-pane/. Accessed: 15 December 2024.

Mendix Docs 2024t. Debug Microflows. URL: https://docs.mendix.com/howto8/monitoring-troubleshooting/debug-microflows/. Accessed: 14 December 2024.

Mendix Docs 2024u. Set Up Error handling. URL: https://docs.mendix.com/howto8/logic-business-rules/set-up-error-handling/. Accessed: 14 December 2024.

Potanski, T. 10 April 2024. What is Mendix and Why to Choose It? Devs_Data. URL: https://devsdata.com/what-is-mendix/. Accessed: 23 November 2024.

Quiroz-Vasquez, C. 16 September 2024. What is debugging? URL: https://www.ibm.com/topics/debugging. Accessed: 14 December 2024.

Rodriguez, N. 29 September 2021. Domain Modeling: What you need to know before coding. Thoughtworks Blogs. URL: https://www.thoughtworks.com/insights/blog/agile-project-management/domain-modeling-what-you-need-to-know-before-coding. Accessed: 23 November 2024.

Salesforce 21 November 2023. Understanding Process Automation: Examples and Benefits. Salesforce ANZ Blog. URL: https://www.salesforce.com/au/blog/process-automation/. Accessed: 9 November 2024.

SAP 2024. What is process automation? URL: https://www.sap.com/products/technology-platform/process-automation/what-is-process-automation.html. Accessed: 9 November 2024.

Schwaber, K. and Sutherland, J. 2020. The 2020 Scrum Guide. URL: https://scrum-guides.org/scrum-guide.html. Accessed: 26 October 2024.

Scrum.org 2024. What is Scrum? URL: https://www.scrum.org/learning-series/what-is-scrum/. Accessed: 26 October 2024.

TIBCO 2024. What is data validation? URL: https://www.tibco.com/glossary/what-is-data-validation. Accessed: 16 November 2024.

Tietoevry 2024a. Tietoevry Sharepoint. Our Company, Company Facts and Materials. Accessed: 11 October 2024.

Tietoevry 2024b. Tietoevry Care. URL: https://www.tietoevry.com/en/care/. Accessed: 17 October 2024.

Tietoevry 2024c. Tietoevry Create. URL: https://www.tietoevry.com/en/create/. Accessed: 17 October 2024.

Tietoevry 2024d. Tietoevry Industry. URL: https://www.tietoevry.com/en/industry/. Accessed: 17 October 2024.

Tietoevry 2024e. Tietoevry Tech Services. URL: https://www.tietoevry.com/en/tech-services/. Accessed: 18 October 2024.

Tietoevry 2024f. Tietoevry Banking. URL: https://www.tietoevry.com/en/banking/. Accessed: 17 October 2024.

Tietoevry Corporation Interim Report 2024. Financial Reports. Tietoevry Q3 2024 English. URL: https://www.tietoevry.com/siteassets/files/investor-relations/2024/tietoevry-q3-2024-english.pdf. Accessed: 11 October 2024.

Tripathy, P. 20 June 2022. 10 tips for everyday Mendix development. Medium. URL: https://medium.com/mendix/10-tips-for-everyday-mendix-development-fb45cb5bb8be. Accessed: 15 December 2024.

Verveg, A. 3 January 2023. A Unified Approach to Workflow Automation & Data Integration. MX Blog. URL: https://www.mendix.com/blog/a-unified-approach-to-workflow-automation-data-integration/. Accessed: 8 December 2024.

Webflow Team 2024. What is UX design? Webflow Blog. URL: https://webflow.com/blog/what-is-ux-design?utm_source=google&utm_medium=search&utm_campaign=SS-GoogleSearch-Non-brand-DynamicSearchAds-Tier4&utm_term=dsa-1480385100845___703207072758___&gad_source=1&gclid=CjwKCAiA9bq6BhAKEiwAH6bqoLW3wP8Cxgno_U9M2z-j-niClrHUsBGJ5ryoPl15jH9dwjL5GDUVshoC8GIQAvD_BwE. Accessed: 7 December 2024.

Appendices

Appendix 1. Terms and Abbreviations

Activity	Action within a workflow
CEO	Chief Executive Officer
CFO	Chief Financial Officer
Constant	Defines configuration values
CSS	Cascading Style Sheets
Extreme	Software development methodology
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
Kanban	Framework in Agile development
Lean	Agile framework based on optimizing resources
MVP	Minimum Viable Product
PoC	Proof of Concept
REST	Representational State Transfer
SCSS	Sassy Cascading Style Sheets
Template	Form used as a guide in making something
UX	User Experience
UI	User Interface
Widget	Reusable component in software application
XPath	XML (Extensible Markup Language) Path Language