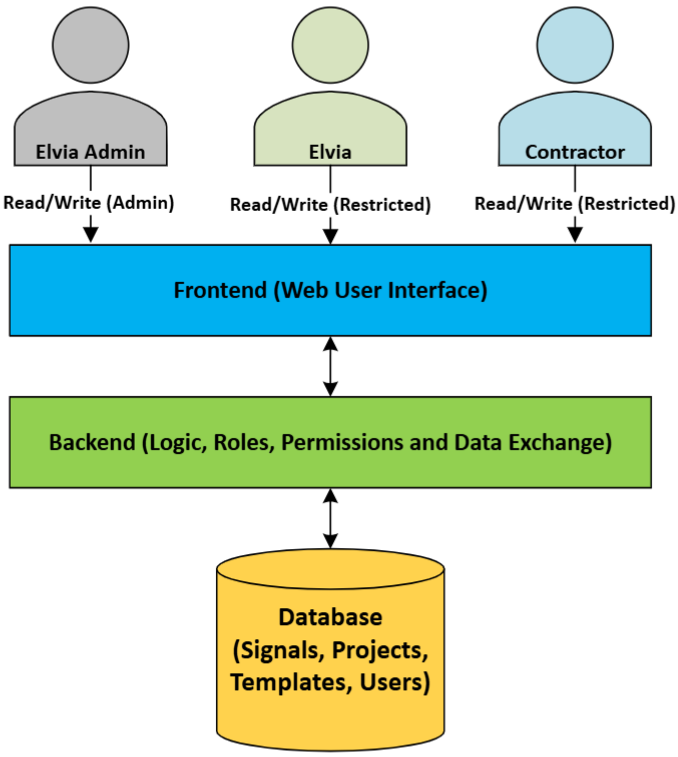
University of South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2026

IT and Automation

A Digital Collaboration Architecture for Multi-Stakeholder Signal Management in Modern Power Grids



Knut Erland A. Strætkvern

University of South-Eastern Norway Elvia

Faculty of Technology, Natural sciences and Maritime Sciences
Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis, 2026

Title: A Digital Collaboration Architecture for Multi-Stakeholder Signal Management in Modern Power Grids

Number of pages: 89

Keywords: Signal List Management, Modern Power Grids, Digital Collaboration Architecture, Multi-Stakeholder

Student: Knut Erland A. Strætkvern

Supervisor: Hans-Petter Halvorsen, Saba Mylvaganam, USN

External partner: Elvia AS

University of South-Eastern Norway

www.usn.no

Summary:

Signal lists are essential in modern power grids. They are used to define the structure, addressing and configuration of signals exchanged between field equipment in the power grid and SCADA systems. Today signal lists are commonly managed using spreadsheets, which introduce significant challenges related to collaboration, version control, traceability and consistency across different stakeholders and projects.

The purpose of this thesis is to explore how a digital collaboration framework can improve signal list management between Elvia and contractors. The system requirements were defined in collaboration with relevant stakeholders at Elvia, focusing on project collaboration, role-based access, signal management and traceability. Based on these requirements a web-based collaboration system was developed.

The system was implemented using ASP.NET Core with Blazor Server, Razer syntax, MudBlazor components in Microsoft Visual Studio and Microsoft SQL Server for data storage. The system was designed to support centralized access, structured data handling and controlled signal management for developing signal lists. Both the system application and SQL database were gradually uploaded to Microsoft Azure cloud platform, making the system accessible online for relevant stakeholders. The system is named "Signal Management Portal – SMP" and can be accessed using this link.

During the project ChatGPT (version 5.3) [1] was used as a supporting tool to improve efficiency in both for report writing and system development. It assisted with report structuring, brainstorming, text formulations and coding-related tasks such as debugging, clarification and support for programming concepts. All AI-assisted content was reviewed and validated by the author and developer before inclusion. AI was not used to making final decisions and conclusions in the report.

The study showed that a web-based collaboration system has the potential to improve consistency, maintainability, version control, and traceability when developing signal lists compared to using spreadsheet-based solutions. The system developed in this thesis confirms the feasibility of the concept and establishes a foundation for further development and a final solution that Elvia can implement in their systems and routines for collaborative management of signal lists in modern power grids.

Preface

This thesis marks the completion of my master's degree in IT and Automation at the University of South-Eastern Norway (USN). The project was carried out by me as a student in collaboration with Elvia AS and USN supervisors Hans-Petter Halvorsen and Saba Mylvaganam.

During the completion of this thesis, I was employed at Elvia, and the project was conducted for the group in which I work. This has been highly valuable, as it has given me first-hand insight into the challenges related to developing signal lists in collaboration with contractors. It has been motivating to contribute to the development of a system that has the potential to solve many of these challenges.

Balancing the thesis work in combination with a full-time position at Elvia has been demanding. However, I am grateful for the opportunity to develop a solution that may support my group and colleagues in their daily work, and this has been a strong source of motivation throughout project.

I would like to express my sincere gratitude to Elvia for providing me with the opportunity to carry out this thesis in collaboration with the company. I would also like to thank my supervisors at USN for their valuable support, guidance and feedback throughout the project, and for encouraging a mindset of continuous improvement and attention to details.

Hamar, 16. May. 2026

Knut Erland A. Strætkvern

Contents

Acronyms and abbreviations

Acronym/ Abbreviation	Description
ACK	Acknowledge - Indication that a signal has been reviewed and confirmed during signal testing.
ADMS	Advanced Distribution Management System – System used to monitor, control and optimize operation of an electrical grid, also referred to as SCADA.
AI	Artificial Intelligence - Technology that simulates human intelligence through learning and problem solving.
DML	Data Manipulation Language – Language used in SQL to manipulate data in the database.
DP	Double Point - Two-bit status indication that is represented by four possible states. The combinations/states are 00-01-10-11.
DTO	Data Transfer Object – Object used to transfer data between different part of a application, like user interface, services and databases.
ER	Entity-Relationship – Describes the tables in a database and how they are related to each other.
IDE	Integrated Development Environment - Software environment to develop, test and debug applications.
IEC 60870-5	Standard for telecontrol communication – Widely used on power system automation for monitoring and control of signals between PAC systems and SCADA.
IEC 61850	Standard for power system automation communication – Widely used in modern substation automation for standardized communication between PAC systems and SCADA.
IOA	Information Object Adress – Numerical identifier that is used for addressing and identification of a spesific signal.
HMI	Human-Machine Interface – Interface that allows operators to monitor, control and interact with systems and equipment locally in a power system facility.
IO List	Input/Output List - Structured documents that describe all input and output signals in an automation system. Used for configuration, testing and maintenance.

PAC system	Protection, Automation and Control System - System that monitors, protects and controls equipment in industrial or power systems.
RTU	Remote Terminal Unit – Device that collects data from a power system facility and transmits it to SCADA, while also receiving and executing remote control commands.
SP	Single Point - One-bit status indication that is represented by two possible states. The combinations/states are 0-1.
UML	Unified Modeling Language – Standardized visual language used to describe and design software systems.

Symbols and units

Symbol	Description	Unit
UA	Unstructured Address - Used in PAC and SCADA systems for signal identification. Calculated using IOA byte addresses.	-

1 Introduction

This chapter introduces the context and purpose of the thesis. Firstly, it describes the background and motivation for the project, then defines the problem and objective before it describes the scope and limitations of the project, lastly it describes the structure of the report.

1.1 Background and motivation

Electrical power systems rely on protection and control systems to ensure safe, reliable operation of electrical infrastructure. The protection, automation and control (PAC) systems are responsible for protection, automation and control functions of infrastructure in the power grid. The PAC systems provide solutions to exchange large volumes of signals that include data from the power grid to centralized remote locations such as the operation center and their integrated systems like SCADA. This makes it possible to access and utilize real-time operational data to monitor, control and optimize the operation of the facility and the power grid. The data exchange is defined using signal lists for individual addressing and specification of the respective signals, their properties and function.

The signal lists serve as an interface between the PAC system and external integrated systems. The list defines which information that is available from the PAC system in the facility for monitoring, control and optimization purposes, and therefore plays a key role in engineering, testing and commissioning stages. When signal lists contain errors, inconsistencies or undocumented changes this can lead to unintended system behavior and reduced or compromised system reliability.

Figure 1.1 shows a conceptual sketch comparing configuration with and without a signal list when mapping signals between PAC systems and SCADA. The arrows in the figure represent signals exchanged across the systems. The purpose of the sketch is to conceptually illustrate the importance of defining and structuring the signals exchanged between the systems. The sketch compares two scenarios with and without a signal list and visualizes how the absence of a structured signal list results in signals not being correctly transmitted between the systems.

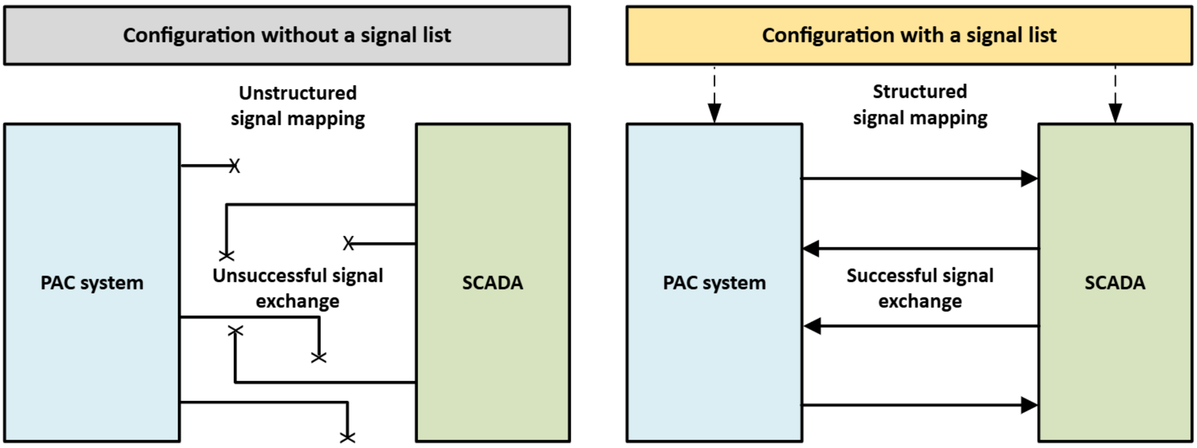


Figure 1.1 - Comparison of configuration approaches without and with a signal list. A signal list provides a structured basis for signal mapping between a PAC system and SCADA. The arrows represent signals that are exchanged between the PAC system and SCADA, and the crosses represent signals that are not successfully transmitted.

In today's practice signal lists are developed in collaboration between Elvia and the engineering contractor for each respective power system facilities. The signal lists are developed using

Excel as the primary engineering tool, and Elvia has developed a generalized signal list template containing all the signals that are relevant for their PAC systems. Elvia and the contractor collaboratively choose which signals that are relevant as this may vary between different facilities. Further the signal list is iteratively developed in Excel using the selected template signals and modified for the specific PAC system for the facility.

While this approach and Excel as an engineering tool offer flexibility and requires only basic user knowledge, it also introduces challenges related to collaboration, version control, maintainability and vulnerability to user errors. Several stakeholders are typically involved in the engineering process of a project, with different levels of experience, preferences and approaches. This increases the risk of inconsistent versions and loss of important changes throughout the work.

Elvia aims to improve their engineering efficiency and data quality by increasing the use of standardized solutions when developing signal lists. To achieve this, there is a need for a new engineering tool that enables a high degree of standardization through templates, signal pools and role-based access control. Project-specific adaptations are of course unavoidable and must be supported by the new solution, although they introduce the risk of inconsistencies across project and remain challenging to manage.

The system proposed in this thesis aims to address these challenges and provide a more effective solution than the current spreadsheet solution. Figure 1.2 illustrates the overall concept of the "Signal Management Portal" system, illustrating how Elvia and contractor users collaboratively can work in the same system through a web-based user interface, which uses the same backed logic and a common database. The solution is designed to ensure that each project contains only one signal list version, providing a single source of truth, avoiding version inconsistencies caused by different copies and versions stored in different locations.

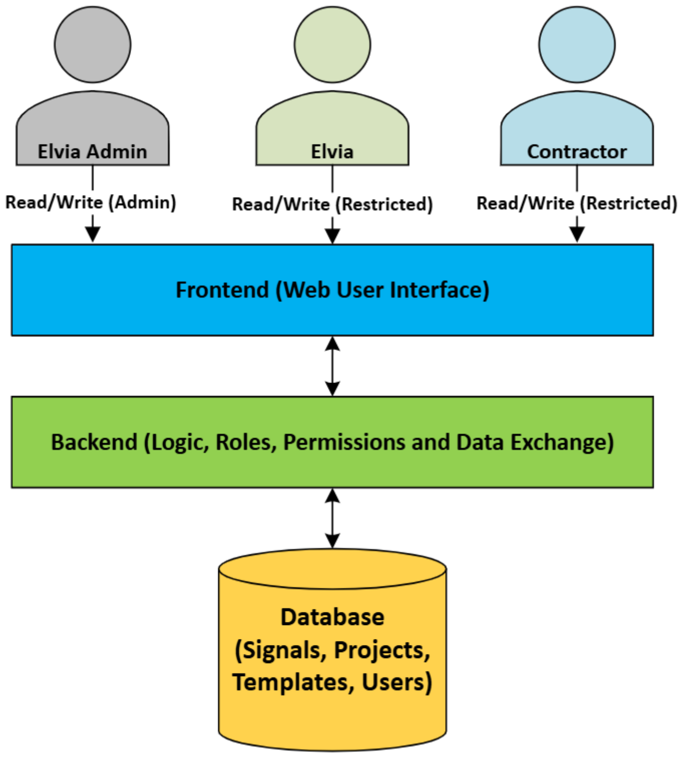


Figure 1.2 – "Signal Management Portal" concept. Showing the architecture of the system at a high level. This figure illustrates how both Elvia and contractor users can access the same web-based user interface, using the same backed logic and a common database for signal list management.

1.2 Problem description and objectives

The goal of the thesis is to serve Elvia with a better tool for collaborative management of signal lists between Elvia and the engineering contractor. Today's Excel-based solution does not provide the functionality expected form a modern system solution. The solution introduces challenges regarding collaborative work, version control and maintainability. This may result in stakeholders working with different, outdated or inconsistent signal lists, thereby increasing the risk of errors and reducing overall data quality.

The current solution also has challenges regarding maintainability of signals and templates. The solution does offer a signal pool and signal templates for different project types, but they are text based and are mostly static information without modern functionality to prevent errors. Any signal modification in the solution must manually be updated in all relevant signals and templates, leading to a high risk of inconsistencies.

Another challenge associated with the current solution relates to signal testing documentation. During the commissioning stage and signal testing between Elvia and the engineering contractors, the signal lists are continuously updated with acknowledgement comments and test results for each signal. Any deviations identified during testing are also documented in the signal list. Due to inconsistencies and limited version control, the signal list must frequently be modified throughout the testing process. As testing is often performed under time pressure, working with large Excel documents introduces the risk of user errors and makes it difficult to maintain an overview of the changes made. Today's spreadsheets often contain large amounts of rows, columns, text and numbers, making them difficult to manage and maintain. As a result, it quickly becomes challenging to work with.

The objective of the thesis is to develop a prototype system for a web-based collaboration framework that aims to solve these challenges. The goal is not to create a fully functioning production-ready system for Elvia, but to demonstrate how a digital collaboration framework can improve signal list management and the engineering process with the contractor.

1.3 Project scope and limitations

The thesis focuses on building a prototype system of a digital collaboration framework for managing signal lists. The solution will not be implemented in Elvia's operational systems as part of this project but can be used as a basis and inspiration for further development.

The scope of the project includes:

- Design and development of a web-based user interface for collaborative signal list management.
- Design and development of backend architecture supporting user roles, access permissions and system management of signals, templates, element types and dropdown items.
- Implementation of key functionalities like signal and template selection, easy modification handling, revision logging, commenting and acknowledging during signal list development and signal testing. The system should also support export signal lists to excel.
- Design and development of a database model for structured storage of signals, templates, system management parameters and revision history.
- Thoughts, evaluation and discussion of potential SCADA integration in future development or solutions.

The scope of the project excludes:

- Integration with Elvia’s operational systems or engineering tools.
- Integration with SCADA for import/export of data.
- Integration with any physical Elvia PAC equipment or network infrastructure.
- Detailed cyber security requirements and assessments.

The limitations are necessary to ensure that the project is feasible within the given timeframe. However, the objective of the project is to design the system so that scalability and future integration are possible.

1.4 Structure of the thesis

This section provides an overview of the structure of the report. The relationship between the main chapters and the overall progression of the work is illustrated in Figure 1.3.

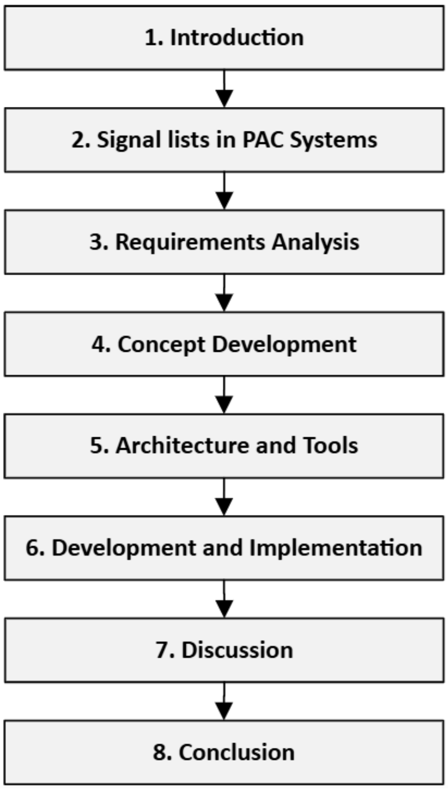


Figure 1.3 - Overview of the report structure. Showing the progression of the main chapters in the report.

2 Signal lists in modern protection, automation and control systems and SCADA

This chapter describes the role of signal lists in protection, automation and control systems and SCADA. Firstly, the chapter explains what a substation is and PAC systems. Then it explains how signal lists are used during the engineering and commissioning stages of a project involving Elvia and engineering contractors. It then describes the purpose of using signal lists in PAC systems and introduces hierarchical structuring and addressing principles and highlights their significance for reliable SCADA integration and operational use.

2.1 Protection, automation and control systems for modern substations in the power grid

To understand the role of signal lists, it is necessary to know the basics of what a substation is and understand the fundamentals of protection, automation and control systems used in modern substations.

A substation is an installation in the electrical power grid, typically characterized by high voltage lines or cables entering a restricted fenced area with large buildings, typically made of concrete. The substation contains primary equipment such as transformers, circuit breakers, disconnectors and busbars, as well as secondary equipment for the PAC system such as protection relays, RTUs, network switches, control and relay panels and HMIs.

The main objective of a substation is to transform electricity between different voltage levels. It serves as a connection point between the transmission and distributional networks. The purpose of the transmission network is to transfer electrical power over long distances with as little energy loss as possible, which is why high voltage levels are used. The distributional network, on the other hand, is designed to distribute and deliver electricity to customers and local communities at lower and more suitable voltage levels [2].

Figure 2.1 shows an example of Tonsen substation which is located in Oslo and is owned and managed by Elvia. The figure shows the high voltage lines (47kV) entering the substation building, and the transformer cells (47-11kV) in the bottom-right corner of the figure. The PAC system for this substation is in a control room, inside the building, behind closed doors, which restricts physical access to authorized personnel only.

Figure 2.1 - Tonsen substation, located in Oslo and owned by Elvia. The figure shows high voltage lines (47kV) entering the building. The grey concrete cells in lower-right part of the figure house the transformers. The transformers transform the voltage to 11kV and deliver power through cables to the distribution network [3].

The PAC systems are crucial for ensuring safe, reliable and efficient operation of substations and the power grid. They combine protection, automation and control functions to monitor high and low voltage grid equipment, detect faults, warnings and alarms, and enable both local and remote operations of substations. Modern substations are equipped with digital and integrated systems that enable communication between electronic devices, control systems, and communication networks. In practice, this allows an operator located in a centralized control center to monitor and control substation equipment remotely using the SCADA system.

Figure 2.2 shows an overview of a modern PAC system for a substation and SCADA, illustrating the connection between substation devices, network infrastructure, SCADA and the signal list. The boxes in the figure describe the main functionality in the different areas. The dashed arrow lines indicate that there is no system integration between the elements, but manual operations, and the solid arrow line indicates that there is a system integration between the elements.

The figure also illustrates how equipment at the bay and station level is connected through station and service network switches. The bay and station levels typically refer to physical or logical divisions within the substation. This structure is relevant to the IEC61850 standard, which is used for design of modern substation automation systems [4]. However, this report does not describe the detailed functionality of this equipment but rather aims to visualize how substation equipment communicates locally inside the substation and to the centralized control center through public network communication infrastructure, like fiber optic cables.

The figure also shows how frontend servers manage the data-exchange of data to the SCADA system, where operators can monitor and control the substation. It also visualizes that the signal list is manually handled by both Elvia and the contractor engineer to configure equipment at station level and SCADA. For simplicity devices such as firewalls, routers and other network devices are not included in the figure.

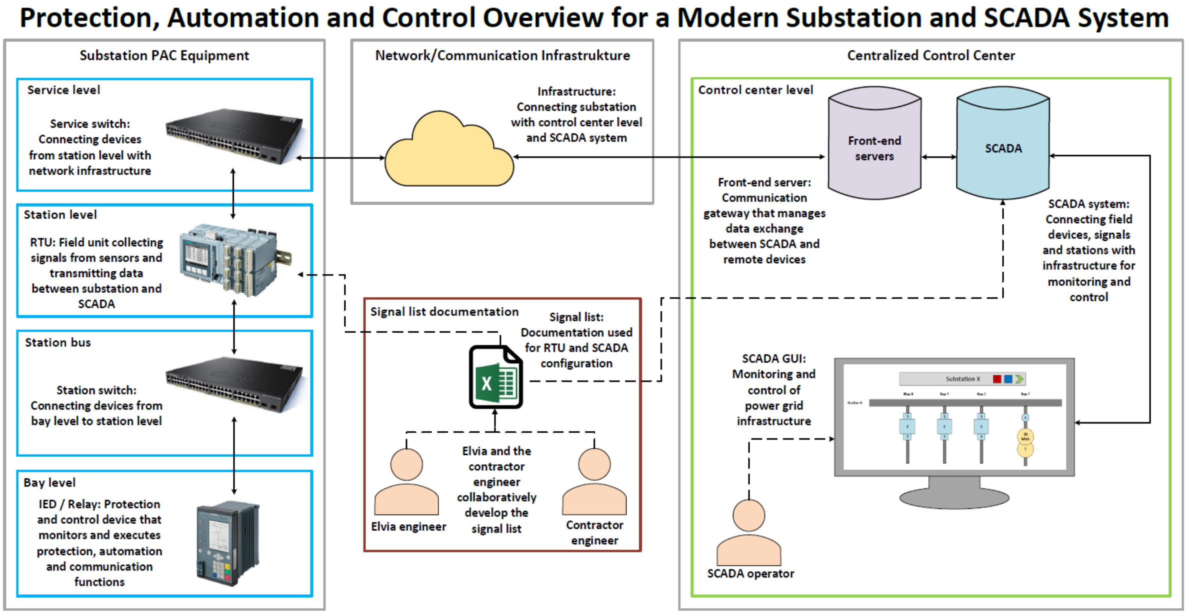


Figure 2.2 – Protection, automation and control overview of a modern substation, illustrating the connections between substation field devices, network infrastructure, SCADA and the signal list.

2.2 Purpose of using signal lists and their relevance in SCADA

Signal lists are essential for protection, automation and control systems. It presents an overview of all the signals that are exchanged between field equipment, PAC systems and supervisory systems like SCADA. The list typically presents a structured specification describing each signal properties and PAC system functionality from a signal point of view. It serves as an interface that connects engineering activities, testing, system integration and documentation for operational use [5]. Signal lists can also be referred to as input-output (IO) lists.

In power system projects that involve PAC systems, the signal lists are developed at an early stage in a project and are continuously updated throughout the project. They first version of the list can be created by either Elvia or the engineering contractor depending on the project type and serve as reference to understanding the signal scope for the PAC system for both parties.

Typically, the signal list forms the basis for import into SCADA. As such, its quality is critical, since errors or inconsistencies in the signal list may result in incorrect structure, signal mapping, increased troubleshooting effort, or reduced reliability in SCADA and the rest of the PAC system. Detecting errors when performing signal testing is frustrating, time-consuming and challenging. Therefore, it is crucial that the content of the signal lists is as accurate and correct as possible throughout the original project, before importing it into SCADA and implementing it into the rest of the PAC system.

2.3 Hierarchical structure of signals in SCADA for building logical substation structures

Signal lists are commonly imported to SCADA and therefore serve as a reference for how signals are organized and parameterized in SCADA. Elvia applies a three-level hierarchy consisting of B1/B2/B3 levels to group signals in their Siemens SCADA system [6].

The hierarchical levels can be compared to a folder structure that categorizes the signals based on their physical or functional purpose. An overview of hierarchical B1/B2/B3 structure for a substation is illustrated in Figure 2.3. Where B1 is the name of the substation, B2 is voltage level and B3 are bays. For example, in Figure 2.3, the signals in SCADA placed under the *Flåklypa – 132kV – T1* structure are signals related to a transformer bay on voltage level 132kV at Flåklypa substation. This organization makes it easier for engineering and operators to navigate in SCADA, particularly when an alarm is triggered for one of these signals, and the operator must handle the alarm.

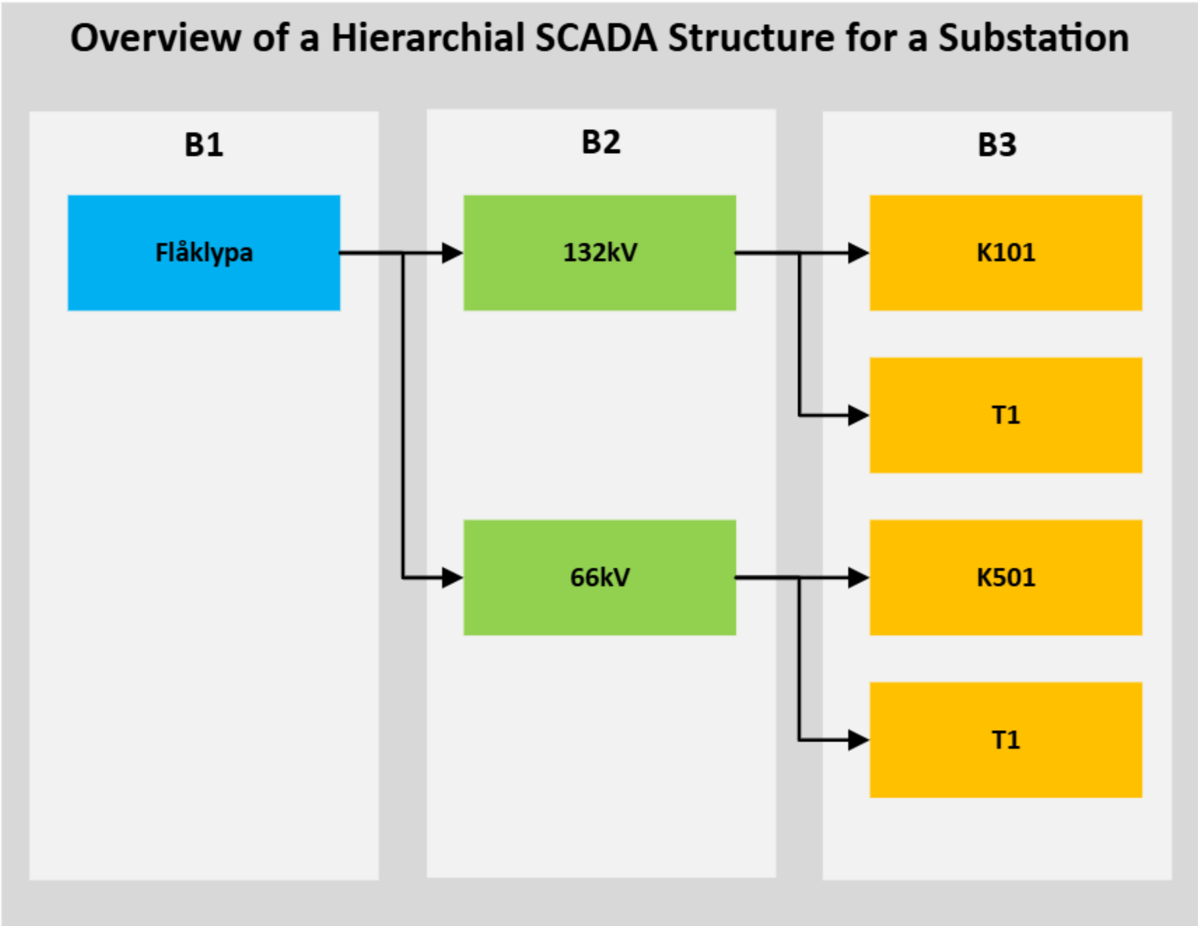


Figure 2.3 - Overview of a hierarchical structure level for SCADA that are used to represent a substation and its facility components. The hierarchy is used to organize the substation into different levels to enable structured navigation, monitoring and control of the electrical equipment in the substation.

The hierarchical levels are typically interpreted as follows:

- **B1:** Station or primary installation
- **B2:** Voltage level or main functional category
- **B3:** Busbar, bay, transformer or secondary functional grouping

A substation generally has one B1 element representing the substation, but typically has multiple B2 elements, and each B2 element typically has multiple B3 elements. The combination of B1/B2/B3 assigns each signal a distinct location in the SCADA system. The hierarchical organization introduces a structured way of organizing signals in SCADA for all substations, resulting in a system that offers good usability and maintainability. Such structure is important in large-scale systems with multiple substations and potentially thousands of signals in operation, because it improves the operator overview, enables easier maintenance and reduces navigation complexity.

2.4 Signal addressing principles for signal identification in PAC systems and SCADA

Signal addressing is fundamental in PAC systems and SCADA as it defines how each signal is uniquely identified and mapped between field equipment, PAC systems and SCADA. A well-engineered and documented addressing scheme ensures that signals can be correctly imported, interpreted and maintained throughout the project and in the operational phase. While poorly structured addressing can lead to signal conflicts, increased troubleshooting effort during testing or maintenance.

The structured and unstructured information object addresses (IOA) addressing approach is a well-established engineering technique that has been used in industrial automation for many years. In IEC 60870-5 based systems the techniques are commonly used to structure the assignment of IOA and are supported by many modern PAC and SCADA systems [7]. The structured addressing approach is described in the IEC 60870-5 standard which specifies the use of three bytes for structuring and defining the signal addressing [8] . The technique divides a signal address into three bytes, which can be used to define structured address ranges. Each byte represents a specific level of the system structure.

The components are typically used as follows:

- **High byte:** Highest logical grouping
- **Middle byte:** Intermediate grouping
- **Low byte:** Individual signal identifier

Some systems and communication interfaces rely on unstructured addresses instead of structured addresses. Therefore, it is common practice in systems to calculate an unstructured address in addition to the structured address. The unstructured address is derived from the High-Middle-Low bytes using a predefined calculation formula typically based on byte-level address allocation. The unstructured address follows Equation 2.1, where the factors $2^{16}$and 2^{8} are used to position the high and middle components in their respective byte position within the address. This results in a combined numerical address consisting of three bytes, where each byte represents 8 bits.

$$UA = (High\ x\ 2^{16}) + (Middle\ x\ 2^{8}) + (Low\ x\ 2^{0}) \qquad (2.1)$$

Example 1 presents an unstructured address (UA) calculation, where the address components are defined as: High = 132, Middle = 1 and Low = 1. The high address is commonly used as a logical grouping related to voltage levels, in this example 132 represents 132kV.

Example 1:

$$UA = (132\ x\ 2^{16}) + (1\ x\ 2^{8}) + 1 \qquad (2.2)$$

By using Equation 2.1 the calculation of Equation 2.2 results in the unstructured address shown in Equation 2.3.

$$UA = 8651009 \qquad (2.3)$$

The example demonstrates the calculation of an unstructured address using Equation 2.1 and 2.2 using three byte components.

Another approach is to derive the High-Middle-Low bytes given an unstructured address. The high byte is derived from the most significant byte, the middle byte from the intermediate byte and the low byte from the least significant byte. The bytes are derived using integer division and modulo operations. Modulo is a mathematical operation that returns the remainder or signed remainder of a division, after one number is divided by another [9].The structured bytes are calculated using Equations 2.4-2.6, where the brackets indicate that the result is rounded down if the result is not an integer. Equations 2.4-2.6 and Example 2 are developed with support from ChatGPT [1].

Equation 2.4 demonstrates the calculation of the high byte by using an unstructured address.

$$High = \left\lfloor \frac{UA}{2^{16}} \right\rfloor \tag{2.4}$$

Equation 2.5 demonstrates the calculation of the middle byte by using an unstructured address and the modulo operation.

$$Middle = \left\lfloor \frac{UA \; mod \; 2^{16}}{2^{8}} \right\rfloor \tag{2.5}$$

Equation 2.6 demonstrates the calculation of the low byte by using an unstructured address and the modulo operation.

$$Low = UA \; mod \; 2^{8} \tag{2.6}$$

Example 2:

Example 2 demonstrates the calculation of a structured address, represented by three bytes by using the same unstructured address that was calculated in Example 1 and presented in Equation 2.3.

Equation 2.7 demonstrates the calculation of the high byte using Equation 2.3 and 2.4. Equation 2.7 shows that the calculated high component decimal number is rounded down to an integer as previously discussed in the chapter.

$$High = \left\lfloor \frac{8651009}{2^{16}} \right\rfloor = \lfloor 132.00390625 \rfloor = 132 \tag{2.7}$$

Equation 2.8 demonstrates the calculation of the middle byte using Equation 2.3 and 2.5. Equation 2.8 demonstrates the use of the modulo operation and that the calculated middle byte decimal number is rounded down to an integer.

$$Middle = \left\lfloor \frac{8651009 \; mod \; 2^{16}}{2^{8}} \right\rfloor = \left\lfloor \frac{257}{256} \right\rfloor \lfloor 1.00390625 \rfloor = 1 \tag{2.8}$$

Equation 2.9 demonstrates the calculation of the low byte using Equation 2.3 and 2.6. Equation 2.9 also demonstrates the use of the modulo operation.

$$Low = 8651009 \; mod \; 2^{8} = 8651009 \; mod \; 8651009 = 1 \tag{2.9}$$

Example 2 demonstrates that the structured address bytes (High = 132, Middle = 1, Low = 1) derived from the unstructured address in Equation 2.3 are identical to the three bytes used in Example 1. This confirms that the structured and unstructured methods are mathematically equivalent and that the conversion between the different two representations is consistent and reversible.

2.5 Signal structure, parameters and element types in signal lists

Beyond hierarchical structure, signal names and addresses, each signal is also defined by a set of attributes that describes signal functionality and how it should be configured in the PAC system and SCADA. These attributes ensure that signals are interpreted and handled consistently across systems for visualization, alarm handling and data logging. A clear and consistent signal structure is therefore essential to achieve correct system behavior, reliable operation and maintainability. While this report does not provide a detailed review of all relevant parameters and attributes used in modern PAC systems and SCADA, it focuses on the overall structure and management of signals and solutions that are used in signal management in SCADA.

Each signal contains a set of parameters that describes its technical and functional properties. Together, all these parameters ensure that signals are consistently structured and correctly addressed across the PAC systems and SCADA. However, maintaining and validating such detailed signal configurations often becomes challenging with hundreds of signals managed using spreadsheet-based tools such as Excel for documentation.

2.5.1 Element types and their role in SCADA and signal lists

To support systematic and structured engineering, signals are parameterized using predefined configurable element types in Elvia's SCADA system. An element type can be understood as a predefined, software configurable component in the Spectrum Power 7 ADMS/SCADA database model that defines a common set of parameters and characteristics shared by multiple signals [10].

By defining element types and assigning signals with different element types in SCADA the consistency is improved, and repetitive configuration work is reduced. Furthermore, updating an element type in SCADA will automatically be reflected to alle signals in facilities that utilize the given element type, this simplifies SCADA maintainability.

Element types typically define SCADA specific attributes. These attributes are typically signal behavior, alarm handling, scaling and graphical visualization properties. The element types ensure standardization across multiple facilities and projects so that solutions become consistent, easier to maintain and provide operators with uniform system behavior and predictable system responses independent of location and across facilities with equivalent functionality.

Table 2.1 shows an example of element type configuration. The “Element type” column represents the configured element types in the system, and the “Element ID” identifies the element type in combination with the element type name. Furthermore, each element type is configured by a set of parameters specified for the specific element type. This report will not describe all SCADA parameters in detail, but only those who are relevant to the project. The “Param X-N” column represents the concept of additional parameters that may be assigned to an element type, but these are kept outside the scope of this thesis.

Table 2.1 - Example of element type configuration used in the SCADA to standardize the configuration parameters and ensure consistent characteristics for similar components and signals.

				Position/state/status text description						
Element ID	**Element type**	**Info type**	**Signal type**	**Intermediate /Middle**	**Off/Out/ Low**	**On/In/ High**	**Incorrect**	**Unit**	**Time delay**	**Param X-N**
1	**A001**	Info	SP		Normal	Alarm				1
2	**V001**	Info	SP		Normal	Triggered				1
3	**M001**	Measurement	SP					A		1
4	**M002**	Measurement	SP					kV		1
5	**E001**	Status	DP	Intermediate	Out	In	Incorrect		On	0

Element type “A001” in Table 2.1, for example, is an information element type configured for alarm-related signals. It is configured for single point signals and therefore has two states: “Normal” and “Alarm”. This element type can for example be suitable for a signal regarding intrusion detection in a substation or oil leakage from a transformer.

While element type “E001” in Table 2.1 is a status element type configured for switch signals with double point indications, such as circuit breakers. This type of element can be used for signals and equipment with multiple operatizing states, like switches.

Using these element types provide SCADA operators with a standardized overview across different facilities. It also enables efficient modification management, ac changes to a specific element type will propagate to all signals using that element type.

2.5.2 Example of a signal documented in an Excel signal list

As previously mentioned in the report, Elvia and the engineering contractor use Excel for signal list documentation in the current solution. The signal list for a modern PAC system may contain hundreds of signals, making the Excel sheet very detailed and challenging to maintain.

Table 2.2 shows an example of a typical generic signal from an Excel signal list illustrating the scope and level of detail for one single signal. As the signal list expands to hundreds of signals, it quickly becomes challenging to maintain, troubleshoot and navigate the list, particularly when most of the content is free text based. Therefore, the need for a more efficient and structured solution is required to ensure consistency, maintainability and improve the useability.

Table 2.2 - Example of a generic signal from a signal list documented in Excel. The signal list contains information of the signals exchanged between the PAC system and SCADA. It contains specific individual signal details and parameters such as hierarchical structure, addressing and parameters.

1/2	Attributes	Text/Value
Transformer T1	**B1:**	Flåklypa
	B2:	132
	B3:	T1
	Type:	DP
	Element:	E
	Element text:	E - Circuit breaker
	Type/source:	IEC61850
	Element type:	E001
	Info type:	Status
	Text Middle/ Intermediate position:	Intermediate
	Text Off/Out/Low:	Out
	Text On/Inn/High:	In
	Text Incorrect position:	Incorrect
	Unit:	-
	Description:	Double point switch breaker
	Signal test comment:	OK – 23.02.2026 - KEAS

2/2	Attributes	Text/Value
Transformer T1	**Time delay:**	-
	RTU name:	Flåklypa
	Monitoring unstructured address:	8651038
	Monitoring Low byte:	30
	Monitoring Middle byte:	1
	Monitoring High byte:	132
	Monitoring type:	31
	Control unstructured address:	8651009
	Control Low byte:	1
	Control Middle byte:	1
	Control High byte:	132
	Control type:	59
	Parameter X-N:	1

This report does not describe all attributes in Table 2.2 but focuses on those relevant to the project. The hierarchical B1/B2/B3 structure described in Chapter 2.3 can be recognized from the table. The structure indicates that the switch breaker signal is associated with Flåklypa substation, 132 kV voltage level and the transformer T1 bay. Further, the signal is identified as a double point signal using element type “E001” in the SCADA, as described in Table 2.1 for a circuit breaker with four possible positions. The description field explains the specific signal, while the signal test comment field is used by the engineers to acknowledge the signal test. The RTU name typically follows the same name as the substation name. Remote Terminal Unit (RTU) will not be described in this report. However, it is a physical component part of the PAC system in the substation which is used for data acquisition and control of the substation.

Depending on the signal type, a signal may include both monitoring and control addresses, using the structured and unstructured formats described in Chapter 2.4. Monitoring addressing represents signals such as indications, faults, alarms, measurements. In Table 2.1 and element type “E001”, the monitoring address represents the switch indication positions of a circuit breaker with four possible positions. While control addressing is used when command signals must be applied from SCADA to the PAC system and substation equipment. This is typically applied for the operation of breakers and regulators, where operators issue commands through

SCADA from the operation center. A typical example requiring control addressing is the transmission of a command to close a circuit breaker to energize a section of the power grid.

The signal in Table 2.2 is defined as a double point signal. This indication encodes four possible states using two-bit combinations. It is applied where it is necessary to distinguish between two end positions of a switching device and to detect intermediate or incorrect states:

- 00 = Middle/Intermediate position
- 01 = Off/Out/Low
- 10 = On/In/High
- 11 = Incorrect position

In practice, double point indications are typically used for switches.

In contrast, a single point signal uses only one bit to represent the state. This is sufficient where only two valid states exist and no intermediate position needs to be represented.

- 0 = Off/Out/Low
- 1 = On/In/High

In practice, single point indications are typically used for faults, alarms and measurements.

Finally, the signal list typically includes several specific parameter settings, such as signal inversion, command duration, characteristic curves etc. These parameters are outside the scope of this thesis and are not described, but the concept is illustrated in Table 2.1 with the "Parameter X-N" column.

3 Requirement analysis for the signal list management system

This chapter presents the requirement analysis for the collaborative signal list management system. The requirements were identified and collected from the project description in Appendix A and from meetings with relevant Elvia resources. The requirements aim to describe how the system should be structured and how it is expected to perform. The FURPS+ model was used to ensure a structured and systematic approach to categorizing the requirements. The model covers functional and non-functional requirements and categorizes them.

The following section is divided into several subsections describing the functional requirements, non-functional requirements, system boundaries and assumptions and data requirements.

3.1 System functional requirements for the signal list management system

This subsection presents the collected functional requirements for the system. These requirements describe the main functionality of the system to support collaborative signal list management between Elvia and the engineering contractor. The functional requirements focus on what the system should do to meet the user's needs. The collected functional requirements for the system are presented in Table 3.1.

Table 3.1 - Functional requirements for the system. The table presents the main functional requirements that the system should satisfy to support collaborative signal list management.

1. The system should provide a digital web-based framework for collaborative management of signal lists between Elvia and the engineering contractor.
2. The web application should have a login page with username/password, with forgotten password functionality.
3. After sign-in, the interface shall display a navigation menu with content depending on the access rights. The menu should at least contain the following pages: • Create new project (Elvia) • Manage existing projects (Elvia/Contractor) • Manage signals, templates, element types and dropdown menu items (Elvia) o Manage signals (Elvia) o Manage templates (Elvia) o Manage element types (Elvia) o Manage dropdown menu items (Elvia) • Mange users (Elvia administrator)
4. The system should provide user authentication and role-based access control. Elvia and the engineering contractor should have different permissions and rights in the system. Elvia

administrators should have full modification rights, while the contractor should be limited to modifying project specific attributes only. Ordinary Elvia users should have the same access as the administrators except user administration control. Signal, template and modifications should be restricted to Elvia users only.
5. The system should have functionality to create ordinary or administrator Elvia users or contractor users. Contractors or regular Elvia users should not be able to create new users.
6. The system should have functionality to add members to a project. If a contractor is assigned a project, it should appear in their interface under their existing projects overview along with all projects for their company. Filtering functions for my projects should be implemented.
7. The system should allow Elvia and the contractor to create signal lists by using predefined templates and manually adding signals from the signal pool defined by Elvia.
8. The system should allow Elvia and the contractor to add signals from a defined signal pool or templates created by Elvia or remove signals if they are not relevant. Every signal in the pool or template must have a unique ID globally in the system.
9. The system should have functionality to create templates with different signals from the signal pool.
10. The system should support project-specific modifications when developing the signal lists. Examples: signal name, IOA addresses, parameters, hierarchical structure (B1/B2/B3). Elvia's existing Excel document template can be used as a guideline for necessary content, but the template should be kept out of the project report.
11. The system should allow Elvia to create, remove, or modify signals in the signal pool or templates. Changes in the signal pool should also be reflected in existing and future templates, but not existing signal list projects.
12. The system should implement a revision log to track changes in the projects to ensure traceability. It should be shown on each signal line in the project.
13. The system should provide functionality for users to acknowledge and comment on signals during signal testing, for both Elvia and the contractor. The acknowledgement should visualize which user and when the signal was acknowledged.
14. The system should provide functionality to export the signal lists to Excel format for offline field use and foundation for SCADA import.
15. The system should provide functionality to display the current project status throughout the project phases. This status can be updated by both Elvia and contractor users.

3.2 System non-functional requirements for the signal list management system

This subsection presents the collected non-functional requirements of the system. These requirements describe system performance and constraints that influence usability, reliability, performance, supportability and limitations.

The collected non-functional usability requirements are presented in Table 3.2 below. These requirements describe the expected quality attributes of the system. This category focuses on ease of use, efficiency and user satisfaction of the system.

Table 3.2 – Non-functional usability requirements for the system. The table presents the usability related requirements that defines how the system is intended to support efficient workflows, system interaction and ease of use.

1. The system should provide a web interface that is intuitive and easy to use for both Elvia and contractor users.
2. The interface should clearly indicate the user's role and permissions associated with the role. The interface should only permit modifications relevant to the user's rights.
3. The system should present the user with notifications if errors, saves, updates, etc.

The collected non-functional reliability requirements are presented in Table 3.3 below. These requirements describe the expected dependability of the system. This category focuses on how the system handles errors, faults and the system stability.

Table 3.3 - Non-functional reliability requirements for the system. The table presents reliability related requirements that define the expected system robustness and stability.

1. The system should provide consistent storage and retrieval of signal data to avoid loss or corruption of the data in the signal lists.
2. The system should update signal pool modifications to the respective templates to ensure that the templates are updated.
3. The system should update element type modifications to the respective signals to ensure that the signal pool signals are updated.
4. The system should update dropdown menu item modifications to relevant areas and elements in the system.
5. The system should avoid version conflicts and corruption by enforcing role-based permissions and controlled data modification.

The collected non-functional performance requirements are presented in Table 3.4 below. These requirements describe the expected system performance. These requirements typically include system response time, scalability and efficiency.

Table 3.4 - Non-functional performance requirements for the system. The table presents the performance requirements that define how the system should perform under normal operating conditions.

1. The system should provide responsive user interactions when working with the signal lists or signal modifications.
2. The system should handle processing of large number of signals in the templates and projects.

The collected non-functional supportability requirements are presented in Table 3.5 below. This category describes how easily the system can be maintained, configured and supported.

Table 3.5 - Non-functional supportability requirements for the system. The table presents supportability related requirements related to system support like maintenance, diagnostics and managing of updates and modifications.

1. The system architecture should be designed using a suitable modern architecture model.
2. The system should be designed using technologies that can be integrated into Elvia systems.
3. The system should be designed to allow for possible future integration with Elvia's SCADA system.

The collected non-functional design challenges and limitation requirements are presented in Table 3.6 below. These requirements define constraints that may restrict or affect the system design and implementation.

Table 3.6 - Non-functional design challenges and limitation requirements for the system. The table presents design related constraints and limitations that must be considered during the development and implementation of the system.

1. The system is created as a prototype to demonstrate a proof-of-concept. The solution will not be implemented in Elvia's systems in the project.
2. The project focus is on the functional requirements, rather than the non-functional requirements to demonstrate prototype functionality.
3. The system will not include any SCADA integration.
4. Cyber security and network infrastructure are kept outside the scope of this project.

3.3 System boundaries and assumptions for the signal list management system

This section presents the boundaries and assumptions for the system. These boundaries contribute to limiting the scope of the system and identify which responsibilities that are handled by the system, and which are not included or considered external. The assumptions are included to clarify and limit the system scope to avoid misunderstandings during the design and development of the prototype system.

Within system scope:

- Web-based interface for collaborative management of signal lists.
- Role-based access control for Elvia and the engineering contractor.
- Signal list management for Elvia and the engineering contractor.
- Signal, system and template modifications are restricted to Elvia users.
- Revision log to ensure traceability of changes in the signal lists.
- Export of signal lists to Excel for offline use.

Outside system scope:

- Integration with Elvia's SCADA and other operational systems.
- Detailed cyber security, network infrastructure and production deployment.
- Developing a mobile-friendly web application.
- Integration with Elvia's existing signal pool or templates. Generic data will be used.

Assumptions:

- Elvia and the contractor must have access to a modern web browser system access during the engineering process of signal list management.
- The signal pool and templates contain generic data for demonstration purposes only.
- The system is implemented as a prototype and is executed for demonstration purposes.

3.4 Data requirements for the signal list management system

This subsection describes the data requirements for the system. The system is required to store and manage potentially large volumes of structured data. The data includes information related to signal list projects, signals, templates, users and revision history. The system should support data consistency and traceability for multiple Elvia and engineering contractor users.

The system should support a centralized signal pool, containing standardized signal definitions by Elvia. Signal lists and templates should reference signals from this signal pool to ensure standardization and consistency and allow for project specific modifications. The project specific modifications should be stored at the project level only and should not result in changes in the defined templates, signal pool or other projects.

The system should store user information to provide authentication and role-based access control. The system should also store revision log to track project and signal list changes.

The system should support export of signal lists to Excel format. The exports should contain all relevant information necessary for engineering and commissioning purposes. The stored data should be suitable for future integration with external Elvia systems such as SCADA. To enable future integration the data must be structured and well defined in the database and in the exports.

4 Concept development and design of the signal management system

This chapter explains how the concept development and design phase of the project was conducted. The phase focused on gathering information and developing an understanding of how signal lists are created, used and maintained, as well as identifying the challenges and limitations if the existing solution. It also involved collecting ideas and suggestions for features and functionality that Elvia wanted to include in the developed system. These inputs were identified by discussions with relevant stakeholders in structured meetings and smaller interviews. The participants had insight into both operational needs of the future solution and limitations with today's current approach.

Based on input from Elvia and iterative development of sketches, the concept progressed towards a web-based collaboration framework, and the collected requirements were updated throughput the process. During the concept development, functional and usability requirements were prioritized to demonstrate key functionality and solution ideas. These requirements were considered essential for developing a system that improves efficiency in signal list management, while also supporting collaboration between multiple stakeholders within a project.

The concept development phase also focused on defining user roles and access rights within the system. The user roles and access rights were designed to reflect differences in responsibilities and levels of involvement throughout a signal list project. Defining a clear separation of access rights was considered important in the thesis to ensure avoidance of unintended modifications to standardized signals, templates, system configurations and user accounts.

The resulting overall concept presents a system solution designed to ensure structured collaboration with clear definition of user roles and rights, maintainability of signals, templates and system configurations, and traceability throughout a signal list project. The concept forms the foundation for system architecture and system implementation.

4.1 Concept sketches of the signal management system web user interface

The initial stage of the development process focused on creating concept sketches and idea drawings for the system using Microsoft Visio. The objective of these illustrations is to provide a quick overview of ideas and functionality. They enabled effective communication with relevant Elvia stakeholders and facilitated early feedback and discussion before progressing with more detailed design and development activities. The remainder of this subchapter presents the sketches developed during the concept phase of the projects. The sketches serve as a basis for the development of the system.

Figure 4.1 shows the concept for the system login page. The original idea was that the user should select their user type, either Elvia or contractor, before signing in with their username/password. But based on feedback from relevant stakeholders, the solution was adjusted so that the system automaticcly identifes the user type based on the user's email address, eliminating the need for manual user type selection. Forgotten password functionality was included to allow users to request a new password.

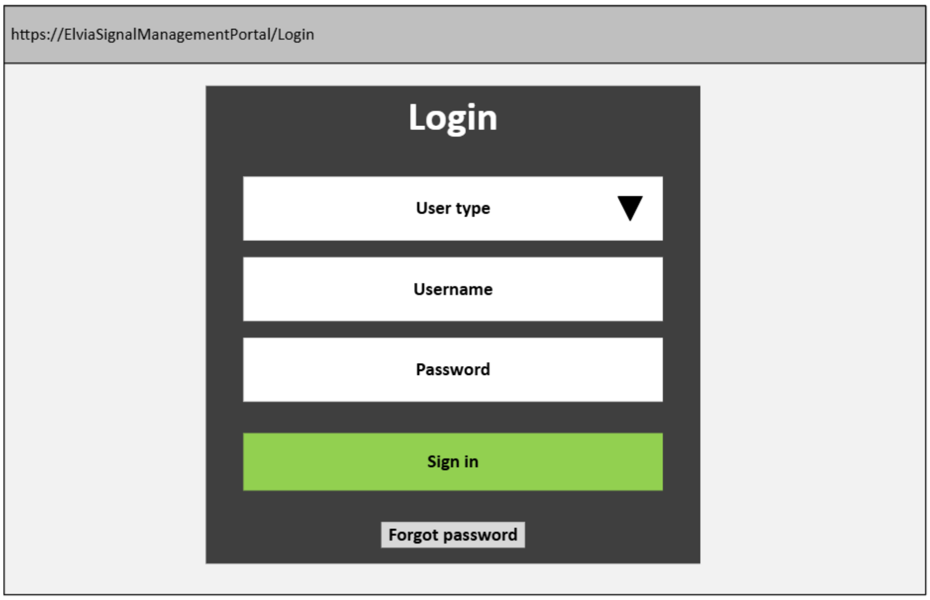


Figure 4.1 – Concept sketch of the "Login" page for the system. It requires the user to enter their username and password to access the system, while user type is automatically identified in the final system. If the user does not remember their password, the forgotten password function can be used to request a new password.

Figure 4.2 shows the concept of the home page for Elvia users. The idea was to organize the different topics in the system into separate boxes, each providing navigation to relevant pages. The system requires access control rights, as only Elvia users should be allowed to administrate signals, templates, element types and dropdown menus. Contractors should only have access to create new projects and viewing their existing projects. The solution was later adjusted by introducing an additional box for user administration, which is only accessible by Elvia administrators.

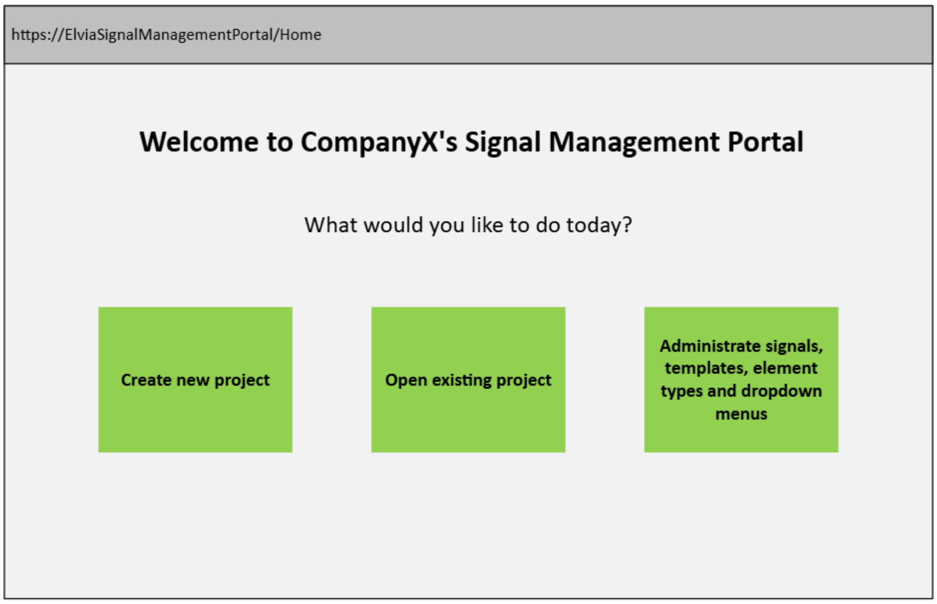


Figure 4.2 – Concept sketch of the "Home" page for Elvia users. This page is displayed after the Elvia user signs in. From here, the user can navigate to different pages depending on their purpose or task. The concept was later adjusted by adding a separate box for user administration, only accessible for Elvia administrators. Contractors are restricted to creating new projects and viewing their existing company projects.

Figure 4.3 shows the concept for the page to create a new signal list project. The idea is that both Elvia and contractor users should be able to perform this operation. Before creating the project, the users must enter key project information and can then continue to the engineering phase, where the signal list content is developed. Later the page was adjusted by separating the members box into separate boxes for Elvia and contractor engineers.

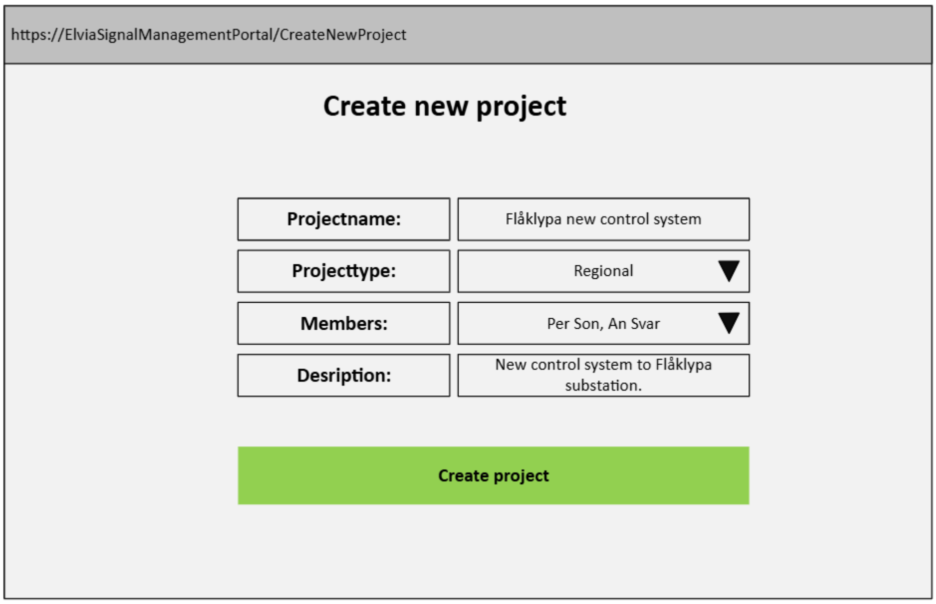


Figure 4.3 – Concept sketch of the "Create new project" page. The page allows the users to create a new signal list project by entering the required project information.

Figure 4.4 shows the concept for the page displayed after the project has been created. This page is where the users perform the engineering work by building the signal list contents. The signal list development should be created by using predefined signal templates and by selecting predefined signals from a signal pool. A revision log solution should be implemented to show which users have made modifications to the signal list. During the development process, this part of the solution was significantly adjusted based on technical considerations, usability needs and stakeholder feedback. In addition, the initial concept sketch does not represent the level of detail required in the final solution.

Additional and more detailed sketches should have been created for both the engineering and commissioning pages for the system, as this would have provided a clearer basis for system development.

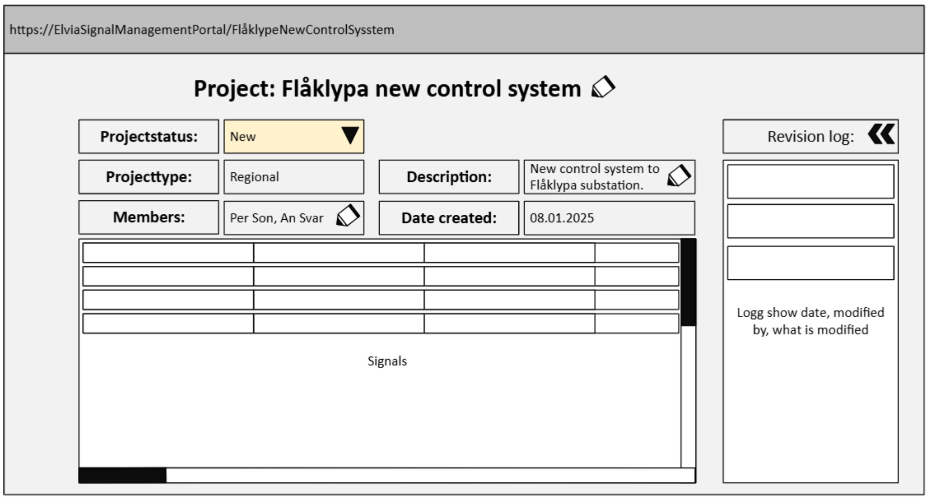


Figure 4.4 – Concept sketch for the "Specific project" page. Illustrating the project details in the top section and the engineering section in the bottom of the figure. The page displays key project information and enables the user to build and configure the signal list content as part of the engineering phase.

Figure 4.5 shows the concept for the existing projects page. The page is intended to provide users with an overview of relevant existing projects and includes filtering functionalities to help the uses to locate projects efficiently. This concept is quite similar to the final solution. However, the final solution also allows users to filter project by project status and includes a button identifying their specific projects.

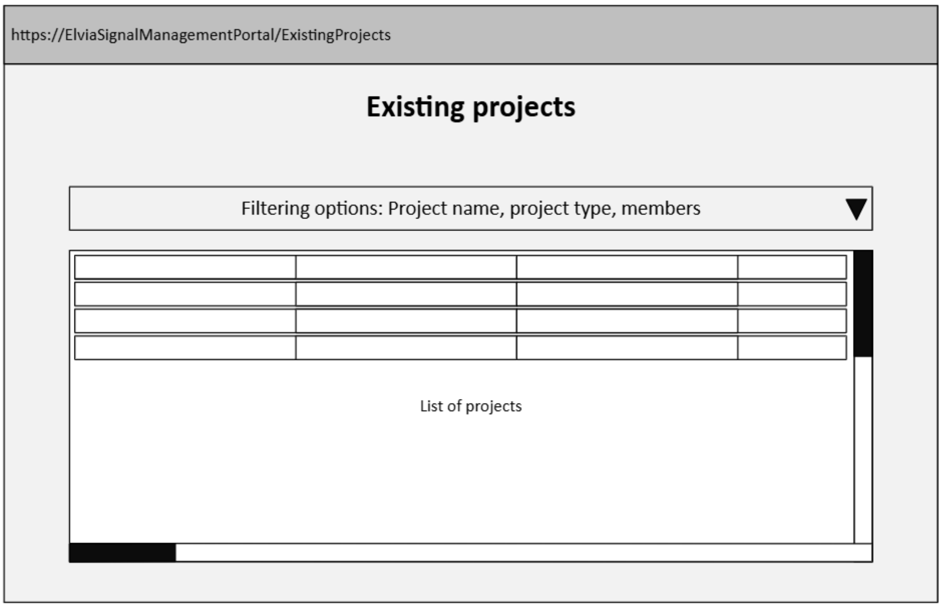


Figure 4.5 - Concept sketch of "Existing projects" page. This page displays the projects created in the signal management portal system. From this page, the user can select an existing project from the list and continue working on it. More functionality was implemented in the final solution for additional filtering options.

Figure 4.6 shows the concept for the administration page for signals, templates, element types and dropdown menu items. This page is intended to be accessible only to Elvia users and provides navigation boxes for the different administration's functions.

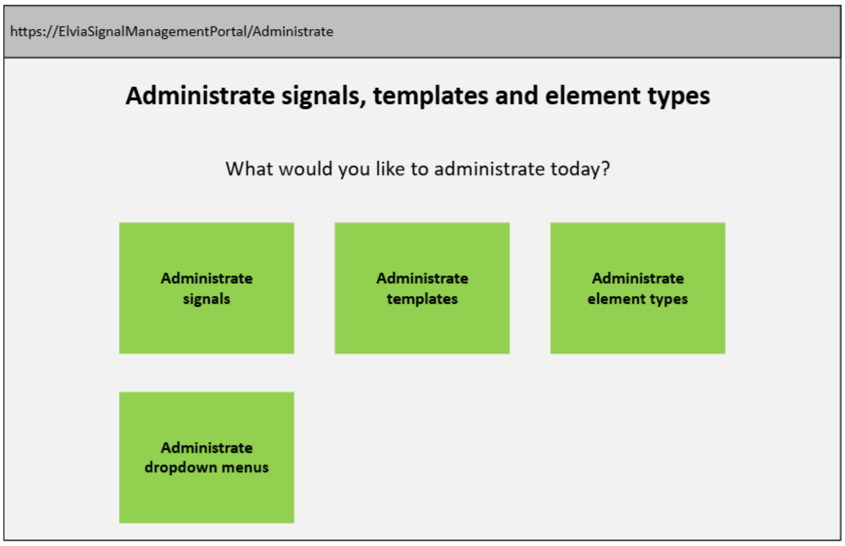


Figure 4.6 – Concept sketch of "Administrate" page for Elvia users. From this page, the user can select which category to perform administration tasks on. This page is only available for Elvia users, not contractors.

Figure 4.7 shows the concept for an administration page. The figure presents a combined view for administartion of signals, templates, element types and dropdown menu items. The idea is that these administration functions should be seperated into four different web pages, but with similar design and structure. The content of each page is decribed in more detailed in Appendices C-F.

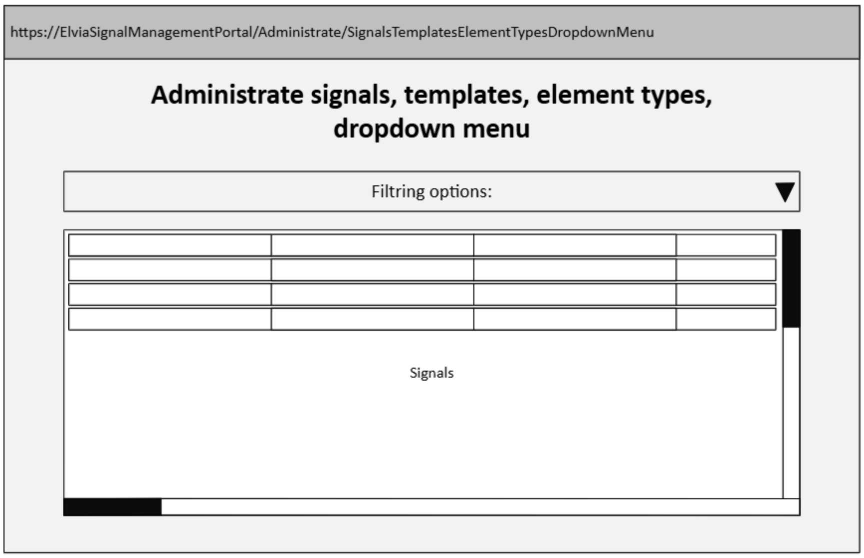


Figure 4.7 – Concept sketch of the "Administration" pages for signals, templates, element types or dropdown menu items. From this page, Elvia users can modify system parameters and manage relevant configuration data.

4.2 Introduction to understanding the use of concept tables and color-coding in the signal management system

This section serves as an introduction to Section 4.3. Its purpose is to explain how to read and interpret the concept tables described in Section 4.3 and in Appendices C-I.

The concept tables in the appendices apply a color-coding scheme where different field types are represented by different colors. In this context, the column header indicates the category associated with the cells in that column.

The color categories are defined as follows:

- **Green**: Text-based fields used for configuration, comments and project specific input.
- **Orange**: Dropdown menus used to support standardized configuration and signal list content.
- **Purple**: Checkbox fields used for acknowledgement. This category may also indicate deviations or signals that are no longer in use depending on the color-coding linked to the deviation options in the dropdown menu.
- **Grey**: Read only fields used for information that is restricted from editing at a given project stage.

The color-coding is implemented in the final system, to provide the users with an overview of necessary input while working with the system.

Figure 4.8 provides an example illustrating the color-coding applied in Appendix H - Signal test management in commissioning mode for Elvia.

In this example:

- The “ACK Elvia” column contains editable checkboxes for acknowledging throughout the signal testing for Elvia users.
- The “Deviation Elvia” column contains dropdown items with predefined deviation categories regarding the signal testing.
- The “Elvia Comment” column contains editable text fields used for commenting registered deviations regarding the signal testing.
- The “User Elvia ACK” column is read-only and displays the user who has acknowledged the testing of the specific signal.

ACK Elvia	Deviation Elvia	Elvia Comment	User Elvia ACK
☐	Fault	SCADA addressing error	
☑			KESR
☐	Remaining	Pending	
☐	Discontinued	Not in relay plan	

Figure 4.8 – Color-coding example illustrating the category principles and signal test status logic in the signal management portal for Elvia users in commissioning mode.

The "ACK" columns also applies to a color-coding scheme to indicate whether a signal has been acknowledged or whether a deviation has been registered. The color represents the following statuses:

- **Blue**: Acknowledged signal; tested and functioning as intended.
- **Yellow**: Remaining signal; not yet tested due to specific circumstances.
- **Red**: Fault signal; not working as intended.
- **Dark grey**: Discontinued signal; signal can be removed from the signal list.

Figure 4.8 illustrates an example where Elvia has registered a deviation during signal testing for the first signal due to a fault related to SCADA addressing. This is indicated by the red background color in the "ACK Elvia" cell, which appears without a checkbox. These status colors are implemented in the final system to support engineers in the signal testing process to easily acknowledge signals and identify those who need corrective action.

4.3 Concept tables illustrating signal list logic and functional design of the signal management system

This section outlines how the system logic and functions related to signal management are intended to be implemented in the signal management portal. However, it does not provide a detailed explanation of each individual function or logic. Instead, this subchapter refers to Section 4.2 for color-coding and applies to Appendices C-I for more detailed descriptions of the functionality. The appendices also illustrate how the system is intended to be visualized and operated for both Elvia and contractor users. The focus of this section is therefore to describe the purpose of each table in the appendices and the functionality it represents in the system.

4.3.1 Dropdown menu configuration

Appendix C presents the concept of the dropdown menu configurator in the system. The configurator is accessed through the administration page shown in Figure 4.6. The purpose of the configurator is to allow Elvia users to manage the content of predefined dropdown menus that are implemented in the system. The goal of the dropdown menus is to limit free-text input and thereby supporting a more standardized system configuration.

The appendix illustrates the different configuration groups and the items associated with them. The categories can be regarded as separates categories without any dependencies between them. The Elvia users can add or remove items in the categories, while changes to the category groups require further system development, as these are integrated and used in the system. The green background color on the column header indicates that the items are entered using free-text input. Adding existing items in a group should be prevented in the solution by displaying a warning message to the user. The appendix also shows user-tracking fields to identify who modified each dropdown item.

4.3.2 Element type configuration

Appendix D presents the concept of the element type configurator in the system. The configurator is accessed through the administration page shown in Figure 4.6. The purpose of the configurator is to allow Elvia users to modify element types that are used in the system. This includes adding, removing or modifying existing element types. Using element types supports standardization, consistency and simplified maintenance, since updates can be made to the specific element types rather than to each individual signal that uses that element type.

The appendix illustrates examples of different element types, each with a unique "Element type ID" specified for each row. Each element type has a set of configuration fields where the function of the element type is defined. The appendix also shows how the dropdown menu is included in the element type configuration illustrated by the orange header cells in the appendix. Adding existing element types should be prevented in the solution by displaying a warning message to the user. The appendix also shows user-tracking fields to identify who modified each element type.

4.3.3 Signal management in the signal pool

Appendix E presents the concept of the signal configuration for signals stored in the signal pool and made available for templates and signal list projects. The configurator is accessed through the administration page shown in Figure 4.6. The signal pool is intended to function as a centralized repository in the SQL database, where signals are predefined and configured before using them in templates and project specific signal lists. Each of the signals in the signal pool is associated with a signal text, an element type and a set of parameters that define its behavior and function within the PAC and SCADA system.

The appendix illustrates how signals may be configured and organized before being reused across different templates and signal list projects. While this report does not go into detail regarding individual parameters used in the configurator, these settings are intended to represent technical properties related to the signal's functionality. Adding existing signals and selecting the same element type should be prevented in the solution by displaying a warning message to the user. The concept also includes fields for tracking of modifications illustrating when a signal was last edited and by which user.

4.3.4 Template configuration

Appendix F presents the concept of template configuration in the system. The configurator is accessed through the administration page shown in Figure 4.6. Predefined templates are commonly used in signal list projects depending on the project type. The templates contain signals for the relevant facilities and can be reused when creating new signal list projects. Using templates in signal list projects supports standardization and efficiency in signal list projects. It also improves consistency and reduces the amount of manual configuration required in a project. Without templates the engineers would have to select each specific signal from the signal pool to the signal list project, potentially hundreds of signals for a project.

The appendix illustrates how templates are structured by combining signals from the signal pool presented in Appendix E. It also shows that the selected signals may be adjusted and adapted to the templates. In addition, the templates include predefined addressing structuring and inherit the configuration data from the signal pool. Adding existing signals and selecting the same element should be prevented in the solution. The concept also includes fields for tracking of modifications illustrating when a signal was last edited and by which user.

4.3.5 Signal list project in engineering mode

Appendix G presents the concept of developing a signal list project for Elvia and contractor users collaboratively. The signal list project can be accessed through the “Existing projects” page in Figure 4.5. This stage represents the engineering phase of the project where the engineers develop the signal list for the facility.

The signal list is initially established by creating a new project as illustrated in Figure 4.3. After creating the project, the engineers are directed to the specific project and can begin the engineering of the signal list content. The engineers then choose templates as described in Section 4.3.4 and Appendix F to form the foundation and starting point of the project and add remaining signals from the signal pool or remove unnecessary signals. The engineers must then customize the signal list content to the specific facility by defining names, hierarchical structuring, element identifiers and configuring the IOA addressing. The objective is to minimize the need for manual modifications at this stage by relying on standardized configurations established in the systems described in the earlier sections.

The signal list contains field for tracking modifications, but also introduces acknowledgement-related columns, which are further illustrated in Appendix H and I. The purpose of these columns is to show which signals have been tested and acknowledged, whether deviations or notifications have been registered, and which user that has performed the relevant action. The acknowledgement fields are read-only in the engineering phase to visualize the signal status but become editable in the commissioning stage described in Section 4.3.6. The acknowledgement fields use the color-coding scheme described in Section 4.2.

4.3.6 Signal list project in commissioning mode

Appendices H and I present the concept for the signal list in commissioning mode for Elvia and contractor users, respectively. This stage represents the phase where signals are tested between the PAC system in the facility and SCADA. The purpose of this view is to support the documentation on whether the test result for each signal is satisfactory. Signal testing is typically performed by both users by testing each specific signal at a time and then acknowledging or registering any deviations.

Successful tests may be acknowledged directly by both the users in each of their views, while unsuccessful tests may be documented through predefined categories for deviations and by writing comments. The acknowledgment fields are used to indicate the status of the signal test through color-coding as described in Section 4.2.

5 System architecture and development tools

The system developed in this project is implemented as a web-based solution to support collaborative signal list management between Elvia and the engineering contractors. It was developed to enable centralized access across organizational boundaries without the need for any local software installation to use the solution.

This chapter describes the system architecture and development tools used for building the system. It introduces the technology stack and describes the architectural design principles and the motivation for selecting specific components. The chapter describes in detail the separation of multiple layers in the system architecture with detailed sketches to visualize the overall system structure and the interaction between the layers and main components.

The tools and technologies for the system were chosen based on their suitability for rapid prototyping and their ability to meet the functional requirements of the system. The non-functional requirements were considered secondary and therefore deprioritized in favor of focusing on core functionality to demonstrate a proof-of-concept for the thesis.

5.1 System technology stack and architecture

The overall system was developed using Visual Studio as the primary integrated development platform (IDE). Visual Studio was selected because it supports full-stack development, debugging features and integration with modern web and backend technologies. It serves the possibility of rapid prototyping and iterative development which is well suitable for the project [11].

Both two- and three- tier architecture was considered during the design phase of the system. A three-tier architecture consists of three separate layers for the presentation, application and data layer as demonstrated in Figure 5.1. This architecture offers a clear separation of responsibilities and is well suited for large scale systems that have multiple users and the need for high scalability requirements and backend API [12].

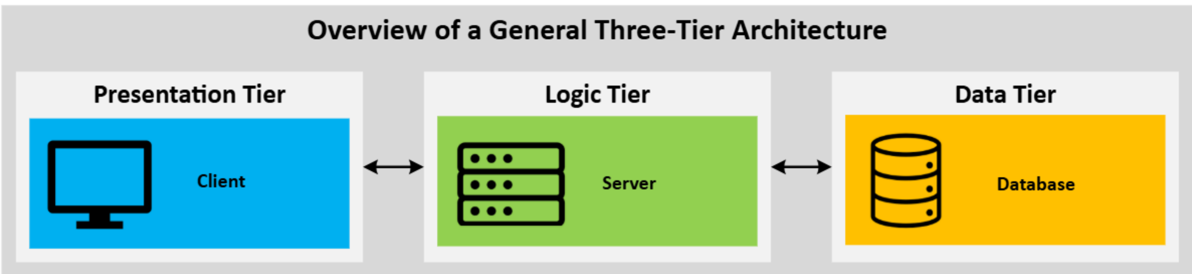


Figure 5.1 - Overview of a three-tier system architecture [13], showing the separation between the presentation, logic and data layer. The presentation tier handles user interaction, the logic tier handles logic and request, and the data tier manages data in the database.

A two-tier architecture was also evaluated. The difference from the three-tier architecture is that the presentation- and logic tier is represented in the same tier, while the data tier is still managed in a separate tier, as demonstrated in Figure 5.2. Using a two-tier architecture reduces the overall development complexity and deployment process, making it well suited for smaller systems and prototype implementations with a restricted number of users.

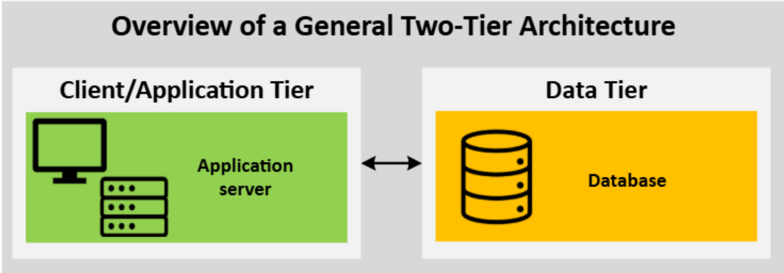


Figure 5.2 - Overview of a two-tier system architecture [13], showing the interaction between the client/application and data layer. The client/application tier handles user interaction and application logic, while the data tier stores and manages data in the database.

Given the scope of the project, limited timeframe and restricted number of users, a two-tier architecture was selected for the system development. This architecture does not restrict production deployment and can be adopted as a sustainable long-term solution for Elvia in further development.

The system was developed using ASP.NET Core, which serves as the underlying web platform handling web requests, hosting and runtime execution. ASP.NET Core provides the core infrastructure required to host the web application, manage requests and execute application logic [14]. The development using ASP.NET Core was greatly inspired by using YouTube tutorials from Hans-Petter Halvorsen's Blog [15]. These tutorials describe how to perform ASP.NET Core web programming and how to connect to an SQL database. Blazor Server was used as an interface framework, enabling both the presentation and application logic to be executed within the same server-side ASP.NET Core application, eliminating the need for a separate backend API. The users interact with the system through a standard web browser, while alle the application logic and data access are processed on the server [16].

The web-based user interface is implemented using Blazor Server components in C#, where Razor syntax is used to define the structure, layout and behavior of the UI components. Razor syntax allows HTML elements and C# logic to be written together when building user interface components [17]. Julio Casal has created a YouTube tutorial for a full beginner's course for Blazor development that was used as inspiration through development. There are also useful YouTube playlists from Microsoft (.dotnet) [18] and Pragim Technologies (Kudvenkat) [19] for Blazor tutorials for beginners that was used as inspiration through development. Microsoft also offers a useful playlist for C# beginner development that was useful during development [20].

To simplify UI development and ensure a consistent visual design, MudBlazor was used as a pre-built UI components library. It offers pre-built, reusable UI components such as tables, forms, and dialogues resulting in a modern, consistent and responsive user experience, that are fully integrated with the Blazor model [21]. During development, YouTube video playlists from ZetBit [22] and Taurius Litvinavicius (Code Exact) [23] [24] were useful to implement MudBlazor components in the system. MudBlazor also has a website with available components ready for implementation [25]. The components are typically visualized with code templates that can be copied and implemented in the system development.

Figure 5.3 illustrates the technology stack for the system, showing how ASP.NET Core, Blazor Server, Razor syntax and MudBlazor are combined to support both the user interface and the application logic.

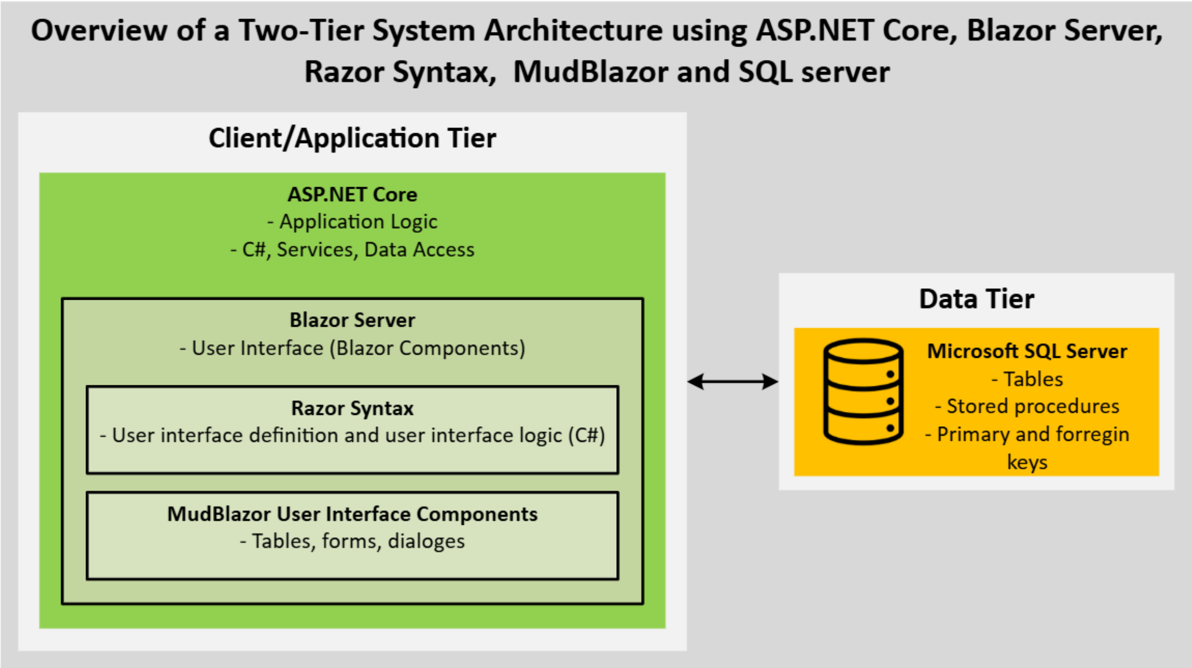


Figure 5.3 - Overview of system two-tier architecture using ASP.NET Core and Blazor server, showing the interaction between the client/application and data layer. The application tier is implemented using ASP.NET Core, while the user interface is implemented using Blazor Server with Razor syntax and MudBlazor UI components.

The SQL database and the web application were first created locally on a computer in the beginning of the development process but eventually was uploaded to Microsoft Azure. Microsoft Azure is a cloud computing platform developed by Microsoft. It offers management, access and development of applications and services to individuals, companies and governments through its global infrastructure [26]. After deploying the database and web application to Azure, the system became globally accessible and independent of the previous local setup. After deployment the SQL database and the web application were able to communicate within the cloud environment. The deployment to Microsoft Azure was inspired by a YouTube tutorial from Hans-Petter Halvorsen describing SQL databases and app services in Microsoft Azure [27].

Further development of the web application was performed locally in Visual Studio and weekly deployed to Azure during the project. Development of the SQL database, however, was made locally using SQL Server Management Studio, but communication directly with the database hosted in Azure. All database modifications were directly implemented in the cloud environment. In future projects, it could be considered to connect the development environment to Azure at an earlier stage, as local development may have been an unnecessary step in this project.

5.2 System data tier domain architecture

The data tier in the system is designed around a version-controlled domain architecture that separates reusable library definitions from project-specific instances. This structure ensures that library modifications and project changes are fully traceable, while preventing library changes to affect existing project configurations. Figure 5.4 illustrates the logical organization of the data tier, the separation of the domains and the flow of versioned data across different system domains.

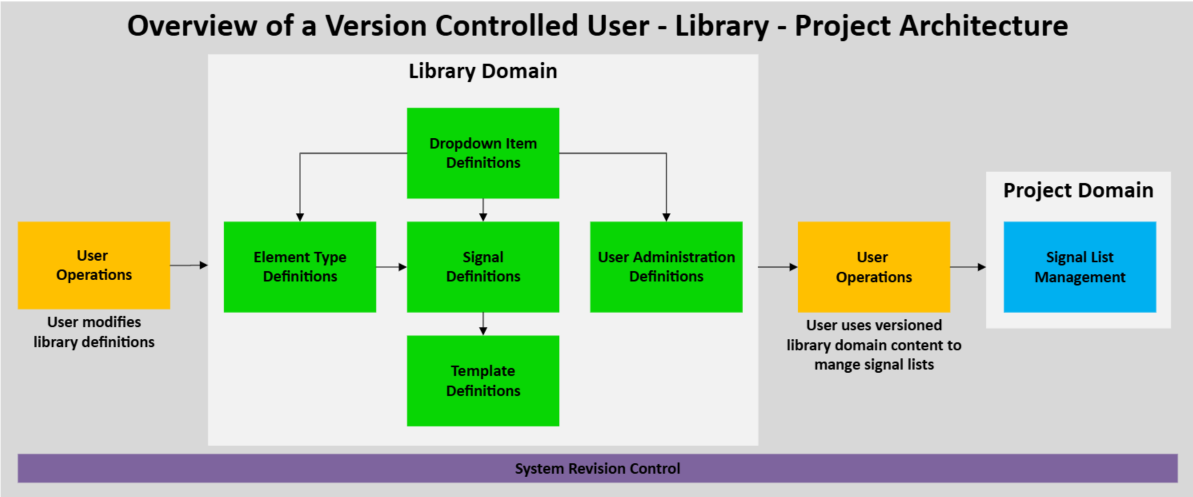


Figure 5.4 - Overview of version-controlled user - library - project architecture. Illustrating how the user interacts with the system through the library domain and how modifications are available to the project domain. It also illustrates how revision control is implemented to ensure tracking of modifications in the system.

Figure 5.4 highlights the need for relational consistency, controlled transactions and revision traceability. These requirements are well supported by Microsoft SQL Server as a relational type of database [28], which was used as the data storage solution in the system. SQL server supports that modifications are applied consistently across the system while maintaining revision logging and traceability [29], as required by the architecture illustrated in Figure 5.4.

MongoDB, as a document-based database [30], was considered as a potential database solution. However, the system requires strict consistency between interconnected elements such as signals, templates and revision logs. Microsoft SQL server provides built-in mechanisms to support these requirements, making it easier to manage and maintain updates and modifications in the system. Achieving the same level of consistency with MongoDB would require additional implementation effort, increasing system complexity [31].

Figure 5.5 illustrates a conceptual comparison between relational and non-relational databases. While relational databases organize data into interconnected entities using relationships enforced by the database, non-relational databases stores data in self-contained documents where interconnection and logic is handled in the application level of the system.

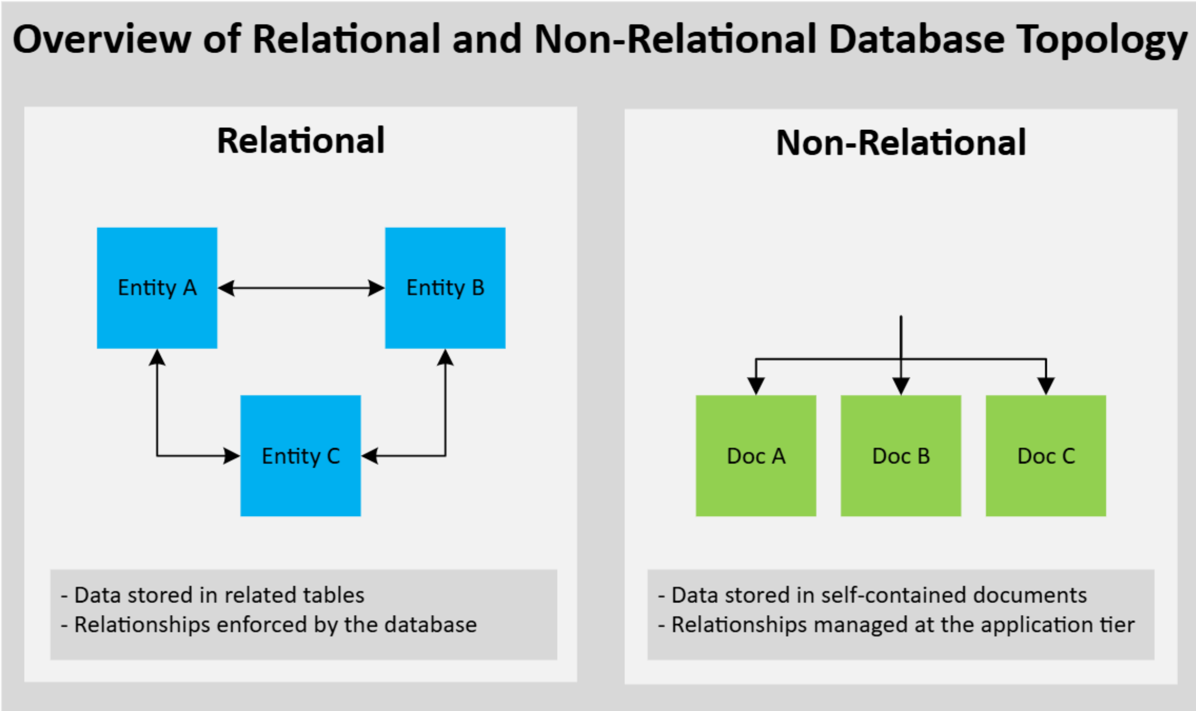


Figure 5.5 - Overview of relational and non-relational database principles [32]. Illustrating the main differences in data structure, design and data storage using relational and non-relational databases.

Since the system required a structured relationship between multiple entities, a relational Microsoft SQL server database was selected for the project.

5.3 Microsoft SQL relational database tables, stored procedures and keys

In relational databases, tables are used to structure and organize data in rows and columns. Each of the tables represents a specific entity within the system and contains columns that define the type of data stored. Examples of typical columns are identifiers, timestamps, parameters. Each row in the table corresponds to individual records containing values for the different attributes [33].

Stored procedures are predefined SQL statements that are stored and perform database operations which enable effective interaction between the signal management portal application and the database. By placing the database logic through stored procedures, efficiency, maintenance and security is improved by restricting direct access to the underlying database table from the application [34].

Database keys are also important to organize and manage relational data. The most used types are primary and foreign keys. The primary key functions as a unique identifier for each row in a table. The objective with primary keys is to prevent duplicate records and ensure reliable data identification. Foreign keys are used to establish connections between tables by referencing primary keys in other tables. These keys help to ensure data consistency between related data in the database stored in different tables [35].

A more detailed and specific description of the relational databases, stored procedures and keys for the “Signal Management Portal” system are described in Chapter 6.

6 Signal management portal system development and implementation

This chapter presents the developed system and its implementation. It describes how the "Signal Management Portal" was developed and realized from the design and concept phase into a functional application. The chapter is divided into three sections. Section 6.1 describes the code design and structure of the application, Section 6.2 describes the SQL database design and Section 6.3 describes the user interface and system functionality, illustrating how users interact with the application.

6.1 Application code design and structure

This section presents the design and structure of the code implemented in the system. The application was developed using Visual Studio and organized using a project structure to support maintainability, scalability and readability. The application was developed using object-oriented programming principles as much as possible, where services/classes, methods and objects were used to separate responsibility and handle different functionalities of the system. The code development was mainly performed using the sources described in Chapter 5, with support from ChatGPT for debugging, troubleshooting, and inspiration during the implementation process. ChatGPT was used as a support tool for development, while the final implementation decisions and evaluations were made by the author.

Appendix J shows a class diagram of the main services, data transfer objects (DTOs), models and classes used in the application, including their relationships and dependencies. A class diagram is a unified modelling language (UML) diagram used to visualize the structure of a system, by showing its classes, attributes, methods and relationships [36]. The class diagram was developed using UMLetino [37], which is a free tool for drawing UML diagrams.

Figure 6.1 shows an example of a class from Appendix J. The class is used for managing signals in the signal pool of the "Signal Management Portal" system. The box contains three sections, where the top section contains the class name, the middle section shows a private variable used inside the class and the bottom section contains public methods that are used by the class and available for other parts of the system. "_dB" is a private variable used in the "SignalPool_Config_Service" class that holds the database connection provider. The methods are used to retrieve all existing signals from the database, updating or inserting signals and deleting signals from the signal pool.

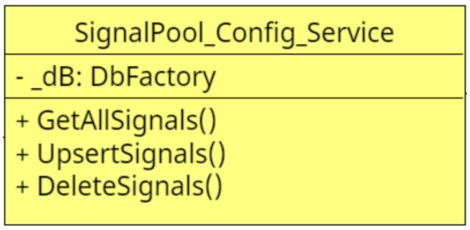


Figure 6.1 – Example of a service class. The figure illustrates a class example of a service for managing signals in the signal pool. The class is divided into three sections, where the top section contains the class name, the middle section the variables and the bottom section the methods for the class.

Figure 6.2 shows a PSEUDO code for managing signals in the signal pool. The PSEUDO code is based on the class shown in Figure 6.1 and illustrates how signals and dropdown values are loaded into the signal pool, how user access is validated and how users can modify the signals. The PSEDUO code also demonstrates how the content is validated before saving the modifications to the SQL database.

```
// PSEUDO CODE FOR MANAGING SIGNAL POOL SIGNALS

START

Load signals from database
Load dropdown values

IF user is not allowed
    Go to Home page

ELSE

    Allow user to:
        Add a new signal
        Edit signal values
        Remove a signal
        Move signals up or down
        Search and filter signals

    When a signal is changed:
        Mark it as changed
        Show unsaved changes warning

    When Save is clicked:
        Check for duplicate signals
        Check that signal text is filled in
        Check that element type is selected

        IF validation fails:
            Show error message
            Stop
        END IF

        Save changed signals to database
        Delete removed signals from database
        Reload signal list
        Show success message

END IF

END
```

Figure 6.2 - PSEUDO code example. The figure illustrates an example of a PSEDUO code for managing signals in the signal pool. The figure describes how signal and dropdown data are loaded and how the user can modify signals.

To provide further context for implementation, the Visual Studio project structure is presented in Appendix K. It provides an overview of how the project is organized into different folders, including Razor components, C# files, layout files, configuration files and other relevant files for the application. The overview also contains comments that describe the purpose and role of each component. The “Models” folder, that contains the application classes, has been excluded from Appendix K, as these classes are already presented in the class diagram in Appendix J.

6.2 SQL database structure and design

This section presents the design and structure of the SQL database used in the system. It explains how the data is organized into tables, how relationships are established between tables using keys and how stored procedures are implemented to interact with the database. Microsoft has published a useful overview of data manipulation language (DML), that is used in SQL server when working with data. It contains an overview of typical DML statements and examples on how to use them - for example, how to modify tables and manage relationships between tables [38]. Hans-Petter Halvorsen also has a blog containing video tutorials supporting these elements [39]. These resources were useful in combination with ChatGPT during development of the SQL database, especially when creating the stored procedures used to manage and retrieve data.

Table 6.1 shows an Entity-relationship (ER) - table illustrating the database tables, their function and their relationships. It visualizes the primary and foreign keys and how they are connected in the database.

Table 6.1 - ER table illustrating tables and keys implemented in the SQL database used to store and manage system data.

Table	Contains	Primary Key	Foreign Key	Related To
Dropdown_Group	Dropdown menu categories that are implemented in the code. These groups/categories contain dropdown items.	Id		
Dropdown_Items	Dropdown items within a dropdown group that are available from a dropdown menu in the system.	Id	GroupId	Dropdown_Group
Element_Type	Contains all configured element types that are modified using the signal manager.	Id	InfoTypeId, PosIncorrectId, PosIntermediateId, PosOffLowOutId, PosOnHighInId, TimeDelayId, UnitId	Dropdown_Item
Engineering_Assets	Contains all assets that are created in the engineering section by Elvia or contractors.	Id	ProjectId	Projects
Engineering_Signals	Contains all signals that are created in the engineering section within different assets.	Id	AssetId, ProjectId	Engineering_Assets, Projects
Password_Reset_Request	Contains all reset password requests from	Id		

	users that have used the forgotten password function.			
Project_Status_Log	Contains all signal list project statuses.	Id		
Projects	Contains all created signal list projects.	Id		
Signalpool	Contains all configured signals that are created in the signal manager.	Id	ControllingTypeId, ElementTypeId, MonitoringTypeId, Parameter1Id	Dropdown_Item, Element_Type
Templates	Contains all configured asset templates that are created in the signal manager.	Id		
Template_Signals	Contains all configured signals within the different asset templates.	Id	TemplateId	Templates
Users	Contains all users with access to the system.	Id		

The database also uses stored procedures to handle data requests in the database. The procedures ensure that data operations like retrieval, insertion and updates are performed consistently and effectively in the database Appendix L provides an overview of the stored procedures implemented in the SQL database.

6.3 Signal management portal user interface and functionality

This section describes the developed “Signal Management Portal” user interface for Elvia and contractor users. The section includes snapshots of the implemented user interface to visualize main functionality and design of the system for both Elvia and contractor users. The snapshots demonstrate how the users interact and navigate throughout the system. The objective is to provide a clear overview of the functionality of the system. However, since not all details and functions are described through snapshots in the report, it is recommended to test the application using the following link to gain a complete understanding of how it works. User accounts can be created by contacting the author or Elvia administrators.

The author has also created a demo project in “Signal Management Portal” called “Master-Demo-Substation – New control system”. The project visualizes a simplified example of a typical signal list project for a substation. Appendix M presents the single line diagram of the substation, and the corresponding assets from the project.

6.3.1 User login and home interface design

This section illustrates the login and home interface for the users. It demonstrates how users can sign-in to the application and the interface that appears after sign-in.

Figure 6.3 shows the login interface for Elvia and contractor users. The users must use their email address as their username and registered password. If the user is not registered in the system an error message will appear showing wrong username or password.

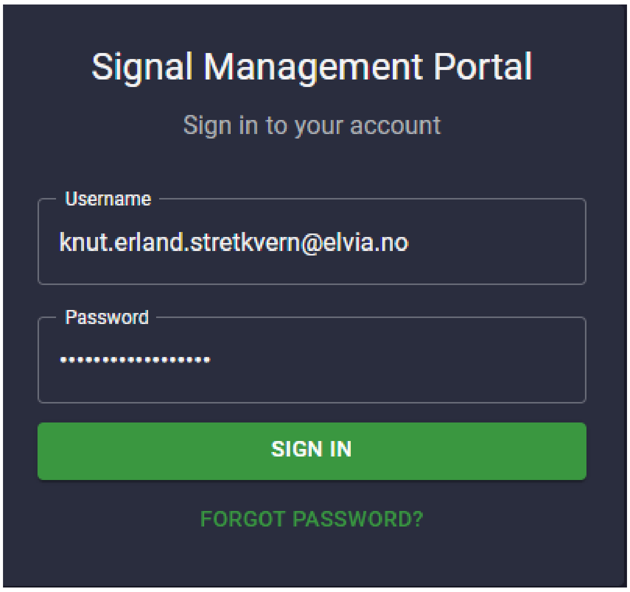


Figure 6.3 – “Login” page for Elvia and contractor users. The users must enter their username and password. The forgotten password function can be used to request a new password.

Figure 6.4 shows the forgot password interface where the user can request a new password. The password request will be sent to the Elvia administrators if the email address is registered in the database. The reason for this, instead of confirming that the user exists, is to improve the security by avoiding confirmation of the user’s existence. This functionality ensures that unauthorized parties cannot verify whether the email address is registered as an actual user in the system.

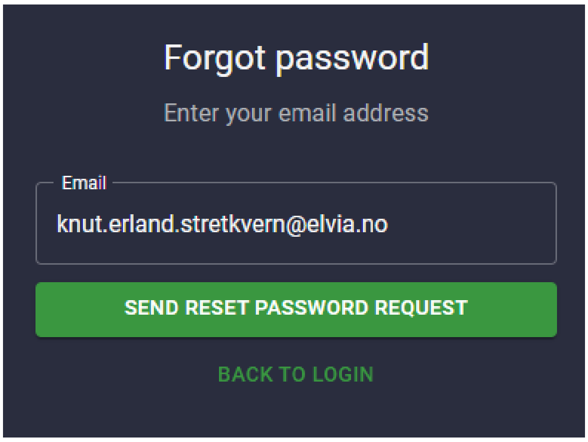


Figure 6.4 – "Forgot password" page. The figure shows the interface after pressing the "Forgot password?" button in Figure 6.3. A request will be sent to the Elvia users if the relevant user account exists.

Figure 6.5 shows the home interface for the Elvia administrator users. This is where the users are navigated after login. The bell symbol on the "Administrate users" box indicates that a user has requested a password change. The "Signal manager" and "Administrate users" boxes are unavailable for contractor users, however, the "Signal manager" box is available for regular Elvia users.

The page has a hamburger menu in the top left corner to navigate to the same pages as diplayed in the boxes in the home page. The users can also press the user account in the top right corner to change their password or sign-out of the application.

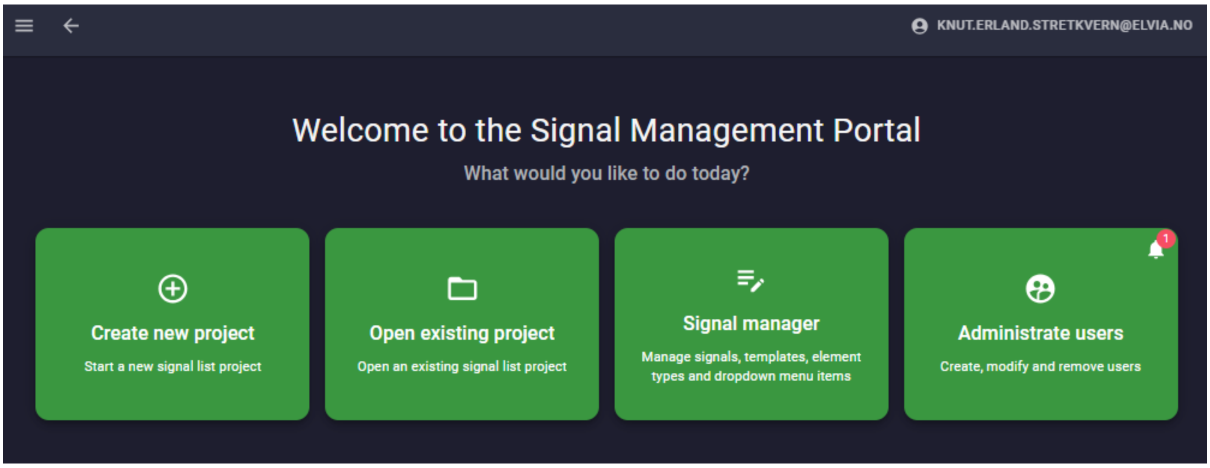


Figure 6.5 – "Home" page for Elvia users. The interface shows the menu for Elvia users after signing in. Here the Elvia user can navigate to other desired pages. The hamburger menu in the left corner can also be used to navigate to other pages. In the top right corner, there is a user account section with user identification and chang password and sign-out functionality.

6.3.2 Creating a new signal list project

This section illustrates how new signal list projects can be created and how existing projects can be modified. Furthermore, it demonstrates how Elvia and contractor users can collaborate when developing a signal list during the engineering process.

Figure 6.6 shows the interface for Elvia and contractor users when creating a new signal list project. The user must enter the project name, project type and specify the Elvia and contractor engineer. The user must also enter a description of the project, and after pressing the "Create project" button the user is navigated to the specific project page, where the user can continue with the signal list development, shown in Figure 6.8.

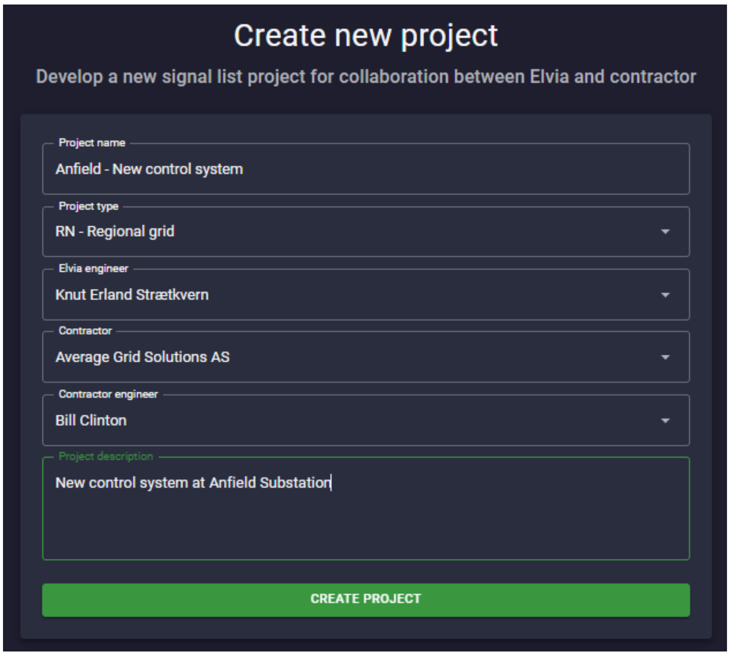


Figure 6.6 – "Create new project" page. Illustrating key project details for creating a new signal list project. The engineer and contractor dropdown menus are retrieved from the user table in the SQL database.

6.3.3 Administration of existing signal list projects, signal list development and signal testing.

Figure 6.7 shows an overview of the existing signal list projects. For Elvia users all registered projects will appear, but for contractors only the projects associated with the company will be available. The users can search for specific projects and use filtering functions to find their desired projects. The overview shows key details of the project, and all the fields are searchable using the search field.

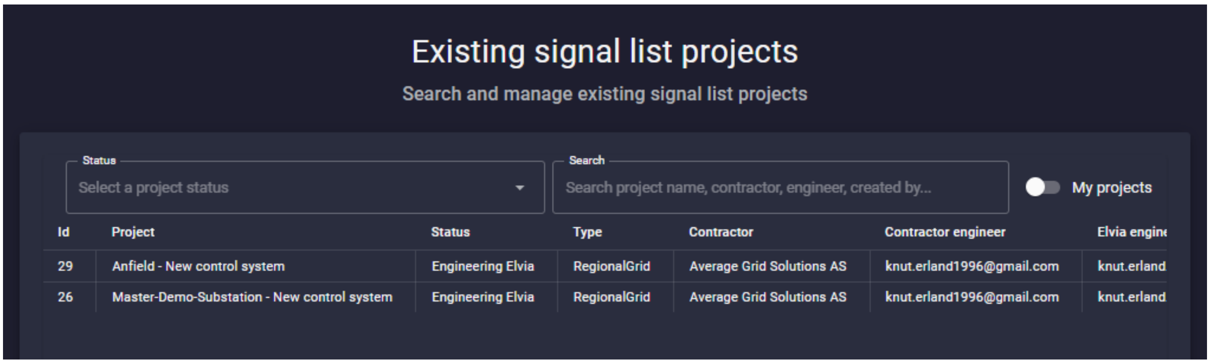


Figure 6.7 – "Existing signal list project" page. Here the users can filter and search for specific signal list projects and select them.

Figure 6.8 shows an overview of a specific signal list project. The red section contains the project details that was filled in when creating the project, as shown in Figure 6.6, and can be view after pressing the "Show project details" button. These project details can also be edited after pressing the "Edit project details" button", and after pressing the button, the window in Figure 6.9 appears. Here the user can update the project details or choose to remove the signal list project. The project detail section also contains a status log box that visualizes the changes in the project status, this can be seen when displaying the project details.

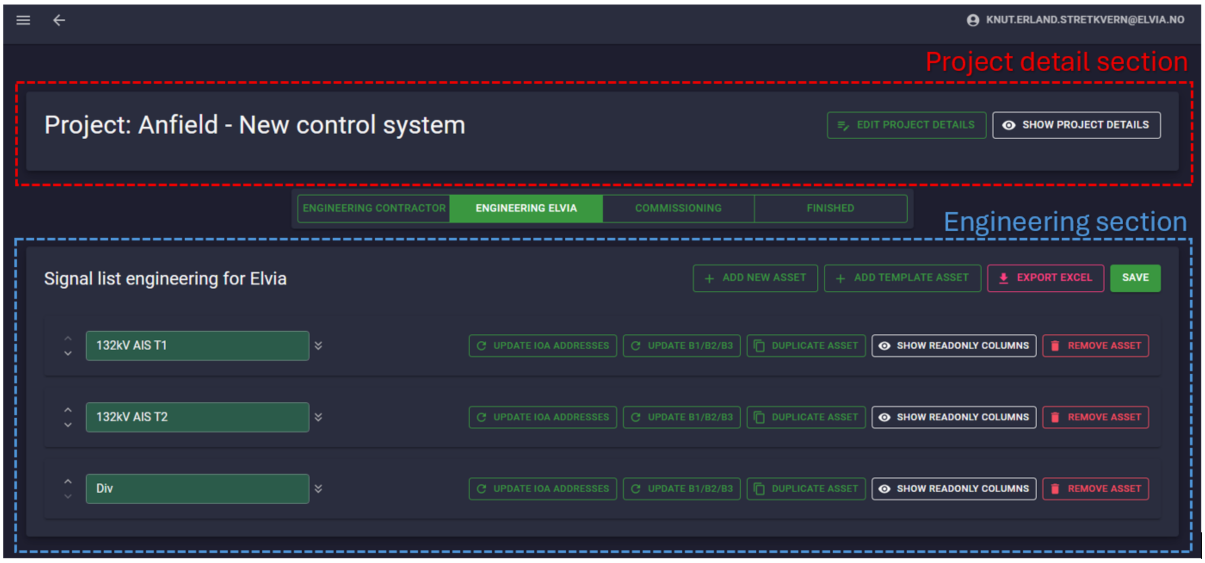


Figure 6.8 – "Specific signal list" project page. Illustrating the page after creating a new project in Figure 6.6 or by opening an existing project. Here the user can view key details regarding the project, change project status and develop the signal list contents.

Between the red project detail section and the blue engineering section there is a project status bar with four buttons. This is where the Elvia and contractor users can change the signal list project status. The first two statuses of the bar decide whether Elvia or the contractor has editing rights to the signal list engineering. The objective is that when Elvia or the contractor user is finished with their part of the engineering project, they should change the status to the other party to review the signal list, and after that they can perform the signal testing by changing the project status to "Commissioning". After completing the signal testing the project status can be changed to "Finished".

Figure 6.9 shows the pop-up dialog for editing the project details of an existing signal list project. The modifications are updated to the database after pressing the "Save" button.

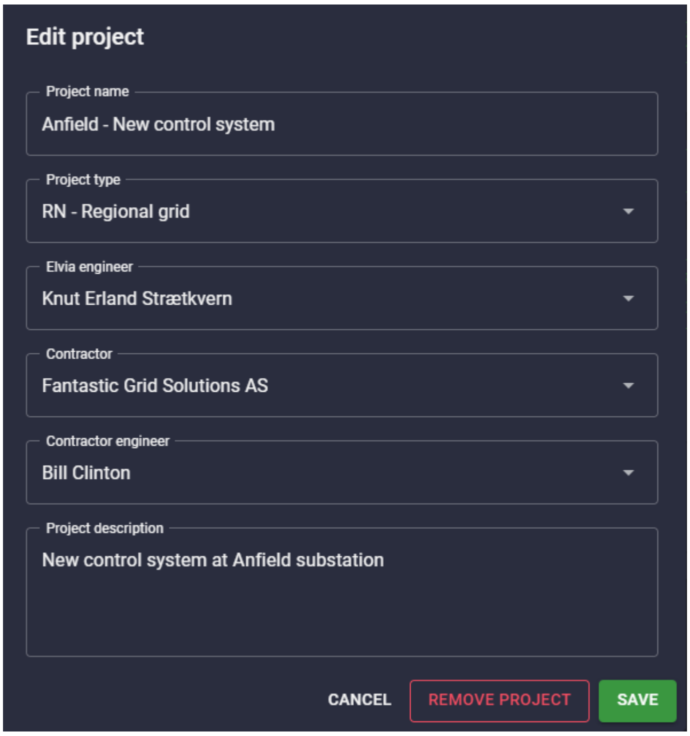


Figure 6.9 – "Edit existing project details" dialog. Illustrating the functionality to edit key project details after pressing the "Edit project details" button in Figure 6-8. The modifications are updated to the database after pressing the "Save" button.

Figure 6.10 shows the signal list engineering view for Elvia users when developing the signal list. This section is where the Elvia users can "Add new asset", "Add template asset" and "Add signal" from the signal pool. Users can also use the "Duplicate asset" button to create an identical asset with identical signals. This function is very useful since many facility assets are similar but vary with different IOA addresses. After so the user must then modify the components specified for the relevant facility.

After creating the assets, the user should first define the B1/B2/B3 structure according to the facility and as described in Section 2.3. It is recommended to use the bulk function to update this structure effectively by pressing the "Update B1/B2/B3" button for each asset, instead of manually filling in each cell in the asset table.

Further the user should continue with the IOA addressing. The system also has a bulk update function by for entering IOA addresses by pressing the "Update IOA addresses" button. This function is useful because typically the high and middle byte of an asset is identical for each signal, while the low address is used for signal identification and must therefore vary.

The system is also implemented with calculation of the unstructured addressing using the formulas described in Section 2.4 and can be shown after pressing the "Show read only columns". The system is also implemented with functionality to check duplicate IOA addressing in the signal list and notify the user if duplicates occur. Furthermore, the system ensures that an element can only be used by one signal per asset. This restriction is included due to SCADA considerations.

The "ACK" columns are read-only columns to show which signals that are tested or have deviations and uses the color-coding as described in Section 4.2. The signal engineering view in Figure 6.10 is identical for contractor users, but depending on the engineering status, the view will be read-only to either of them.

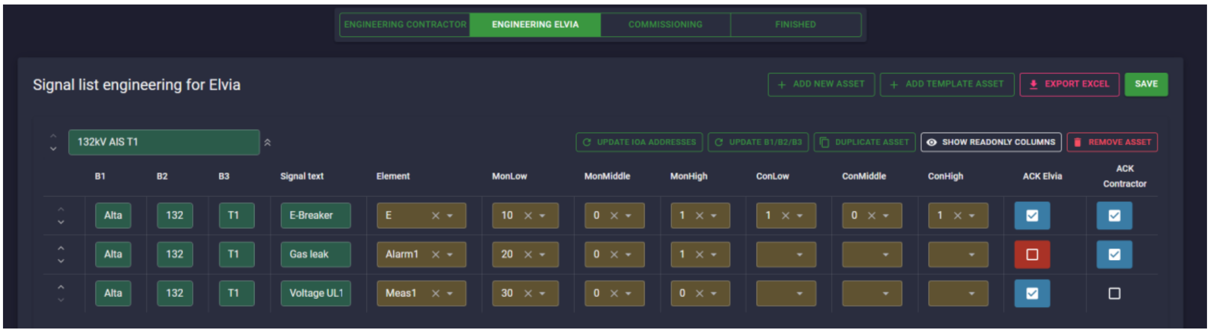


Figure 6.10 – "Engineering view" for Elvia. Illustrating how the engineer can configure the signal list by creating assets and selecting signals from the predefined signal pool. The view is similar for the contractors if the project is in "Engineering contractor" status.

Figure 6.11 shows the view for Elvia users when the project is in "Engineering contractor" status. The figure illustrates that the contents is read-only and changing the project status to "Engineering Elvia" is required to make modifications to the signal list.

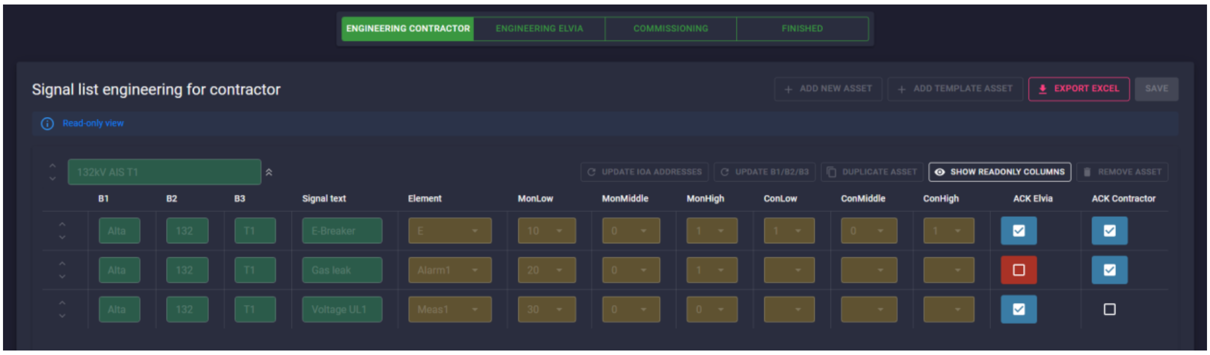


Figure 6.11 – "Engineering view" for Elvia. Illustrating that when the project is in "Engineering contractor" status the Elvia engineer is restricted to having read-only access to the signal list. The view is the same for the contractors if the project status is in "Engineering Elvia".

Figure 6.12 shows the commissioning view for Elvia users. This is where the users collaboratively perform the signal testing, which involves acknowledging or registering deviations when going through and testing each signal from signal list. The view is similar for the contractor users, but they have their own separate columns for acknowledging, deviations and comments, since these may differ from Elvia deviations or comments.

The goal of the signal test is that all operative signals are acknowledged by both parties. The commissioning view also shows which users that have acknowledged during the signal testing and which date, which is useful for documentation purposes. The view also shows how many signals the signal list contains, and how many that have been acknowledged by both parties.

The system also has a function to export the signal list to Excel format using the "Export Excel" button. This function can be useful if the users require the signal list in paper format. The functionality is inspired by a YouTube video from Tarun Saini (ASP.NET MVC) [40].

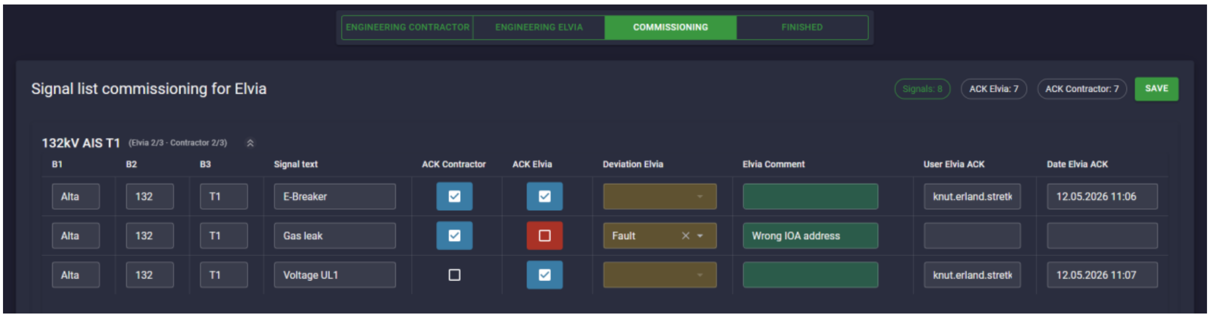


Figure 6.12 – "Commissioning view "for Elvia. Illustrating the signal test interface when the project is in "Commissioning" state. Here the users can acknowledge signals or register deviations. The view is the same for the contractor users, but separate fields for acknowledging, deviations and comments.

Figure 6.13 shows the view when the project is in "Finished" status. This view presents the users with a summary of the signal testing. It displays which assets that are fully tested and which have deviations. Any deviations are listed in the bottom of the view with the comments from the Elvia or contractor users. The view is similar for both Elvia and contractor users.

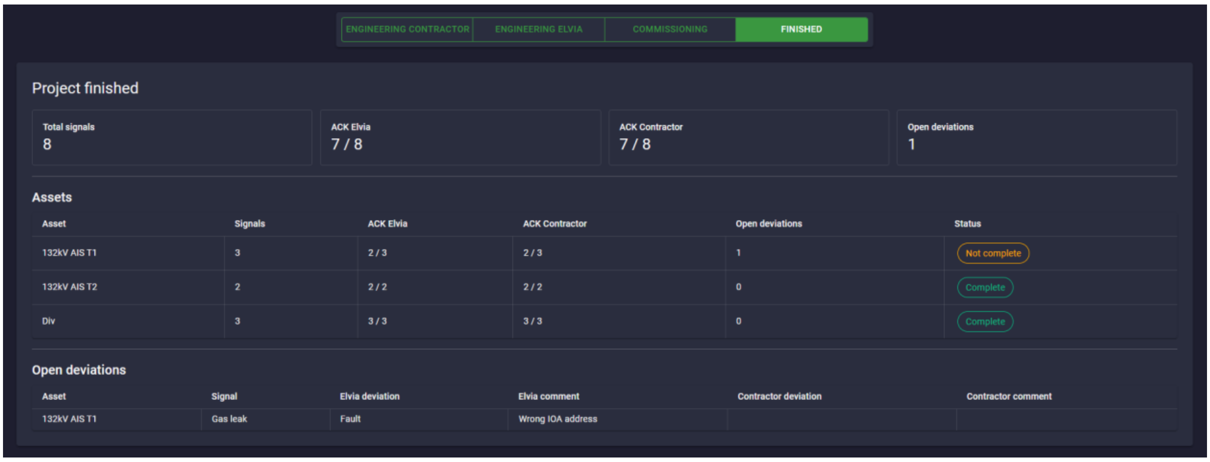


Figure 6.13 – "Finished" view for Elvia and contractor users. Illustrating the signal testing summary with an overview of assets that are successfully tested, and which have registered deviations.

6.3.4 Administration of new and existing system users

This section focuses on user management in the application describing how the Elvia administrators can create new users or modify existing users.

Figure 6.14 shows the "Administrate users" page for Elvia administrators. Here the Elvia administrators can create new Elvia or contractor users using the "Create new user" button or modify existing users. The menu is restricted to Elvia administrator users only, since ordinary Elvia and contractor users should not be able to modify other users. Further the administrator can search for all system users and filter on user type. The view also illustrates which users have requested a new password, indicated with the red bell symbol.

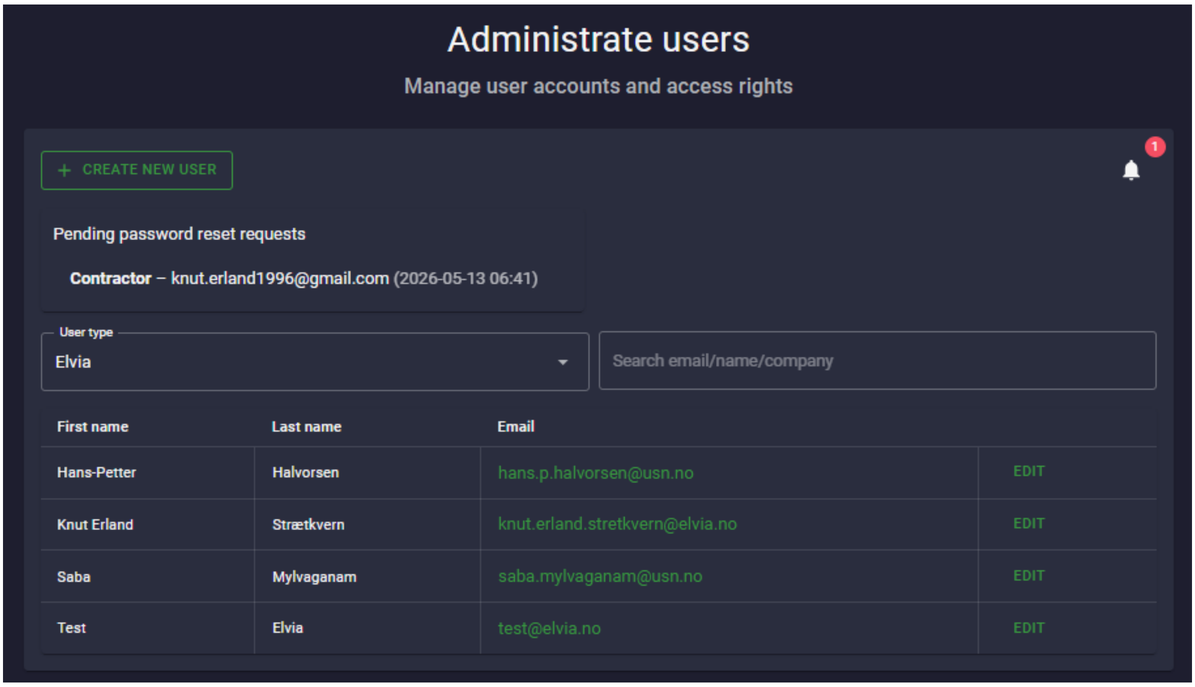


Figure 6.14 – "Administrate users" page for Elvia administrators. Illustrating the interface for modifying existing users or creating new users. Here the administrators can search for existing users and view which users that have requested a password change.

Figure 6.15 shows the "Edit user" dialog after pressing the "Edit" button in Figure 6.14 on a user. This is where the Elvia administrator can edit user details, manage access rights, delete users or generate new passwords, and update the details to the SQL database.

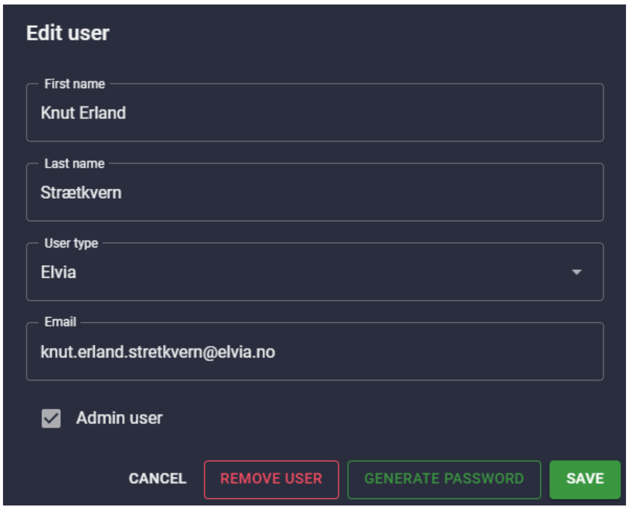


Figure 6.15 – “Edit existing user” dialog. Illustrating the interface for the Elvia administrators after pressing the “Edit” button for a user account in Figure 6.14. Here the administrators can remove the user account or generate a new password. A notification will appear after pressing the “Remove user” or “Generate password” button informing that the action will be triggered when pressing the “Save” button. The generated password is then automatically sent to the user that requested the password change.

Figure 6.16 shows a prefilled email with a new generated password that is automatically sent to the user after the administrator generates a new password. The reason that the link in the email looks unusual it that it has been rewritten by the company’s mail security system and is verified as a safe link.

The user must then create a new password after the first login when using the generated password. The current system solution uses a defined gmail account for sending system emails, but this can be changed to any desired company email account if desired in the application settings.

The solution for automaticly sending emails are inspired by a YouTube video describing how to send emails using .NET [41]. This email functionality is also implemented for sending emails when creating new users and deleting users. The current setup allows the users to reply to this email account if they have anu questions regarding the email. However, some companies may prefer a no-reply email address, so this can also be considered as a possbile solution in future work.

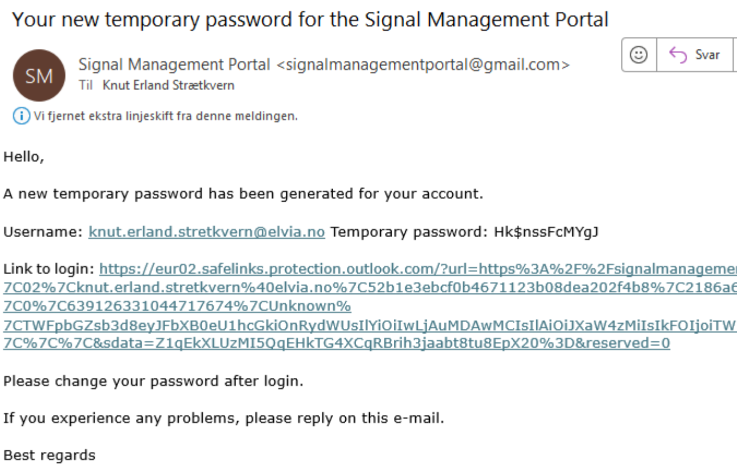


Figure 6.16 - Generated email for password reset. After generating a new password and clicking the “Save” button in Figure 6.15, an email will automatically be sent to the users with a new generated password.

Figure 6.17 shows a prefilled email with a notification of account removal. The notification is automatically sent to the relevant user after the account is removed by an Elvia administrator by pressing the “Remove user” button in Figure 6.15.

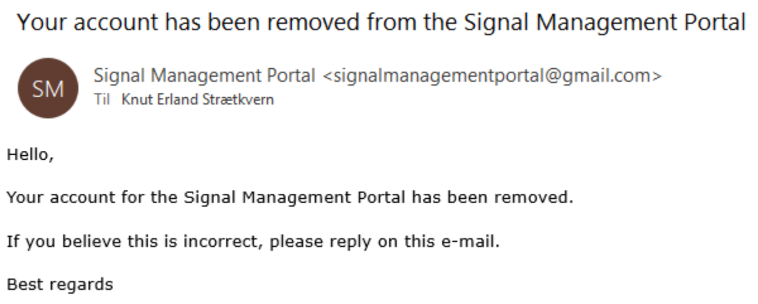


Figure 6.17 - Generated email for user removal. After clicking the “Remove user” button in Figure 6.15, an email is automatically sent to the relevant user notifying that the account is removed.

Figure 6.18 shows the dialog window for Elvia administrators when creating a new system user. The administrator can choose between either Elvia or contractor user type. If the email address contains “@elvia.no” or “@usn.no” the “Company” field will be hidden, as storing this information is not relevant for these user types since they will already be categorized as Elvia users. However, this information is relevant for the contractor users since this is necessary information to define which projects the contractors can access in the system.

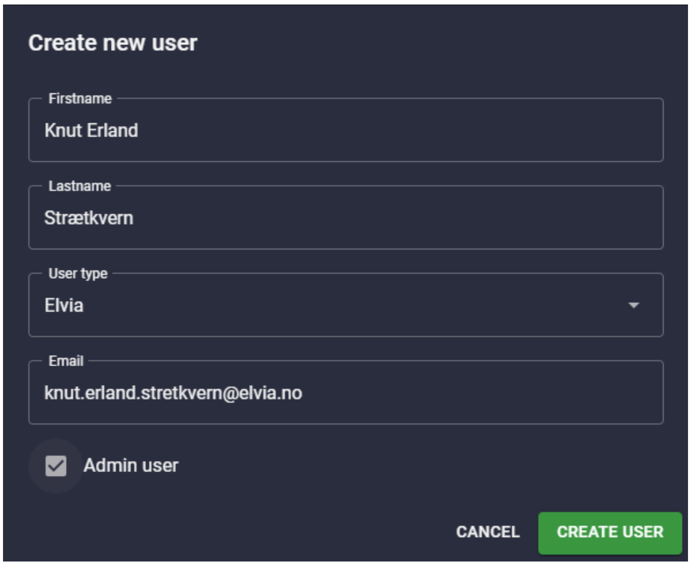


Figure 6.18 – “Create new user” dialog for Elvia administrators. Illustrating the interface when creating a new system user. It requires personalia, email and user type. If the selected user type is “Contractor” the “Company” filed must be specified using a predefined dropdown menu with available companies. If the new Elvia user should have administrator rights the “Admin user” checkbox must be filled.

Figure 6.19 shows a prefilled email with a user sign-in details after creating a new account that is automaticlly sent to the new user. The password is automatically generated by the system. The username and password are then updated in the SQL database, and the password is hashed so it cannot be displayed to anyone. Hashing means converting data, like passwords, into a fixed coded value, so the original data cannot easily be recovered [42]. Functionality for hashing was inspired by a YouTube video from Claudio Bernasconi [43] and a NuGet package description source [44].

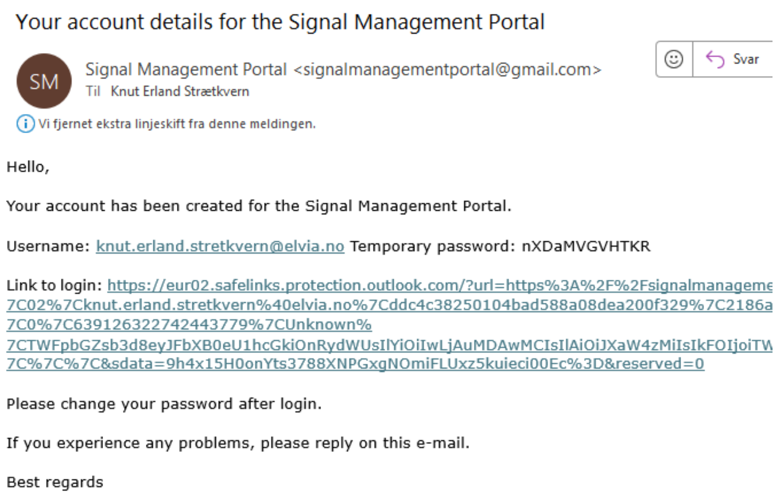


Figure 6.19 – Generated email with sign-in details for a new user. It is automatically sent to the new user after the Elvia administrator presses the “Create user” button in Figure 6.18.

Figure 6.20 shows the interface for changing password. This interface will appear the first time a new user tries to log on using the generated password. The interface will also appear if a user has requested and retrieved a new password from an Elvia administrator or if a user tries to change their own password. The password requirements are listed on the right side, and the checkboxes will update when the new password meets the requirements. The new password will be updated in the SQL database after pressing the change password button. If the users try to reuse the same existing password, an error message will appear indicating that a new password is required. After changing the password, the user is navigated to the “Home” page.

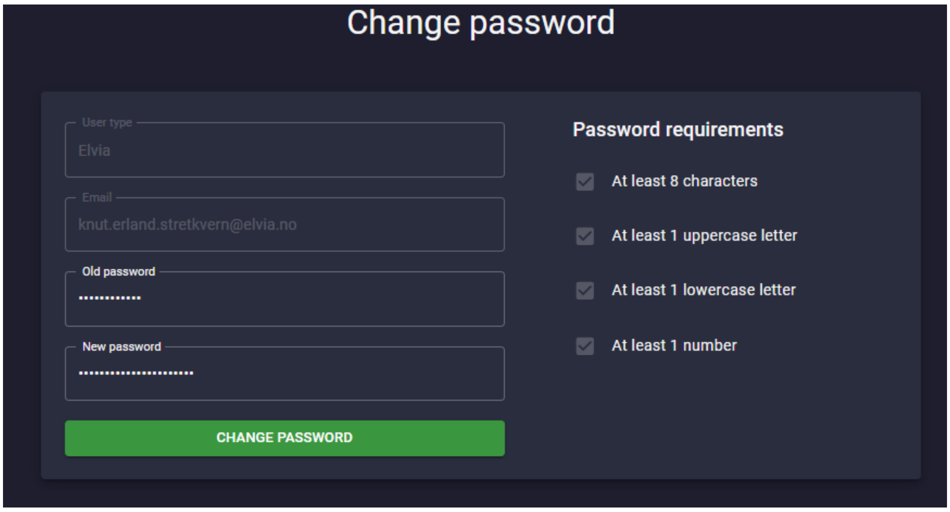


Figure 6.20 – “Change password” page. Illustrating the interface after sign-in using a generated password. The user must set a new password that fulfills the requirements listed in the right section in the figure. The checkboxes are continuously updated when entering a new password, and all must be checked to be able to press the “Change password" button.

6.3.5 System administration of signal pool signals, templates, element types and dropdown menu items

This section focuses on the system administration of signals, templates, element types and dropdown menu items in the application. It provides an overview of the different pages in the "Signal manager" section and explains how they can be used to modify the application content.

Figure 6.21 shows the "Signal manager" page for the Elvia administrators. This is where they can navigate to other administration pages for managing the signal pool signals, templates, element types and dropdown menu items. The "Signal manager" section is restricted to Elvia users only. Contractors do not have access to this section in the application.

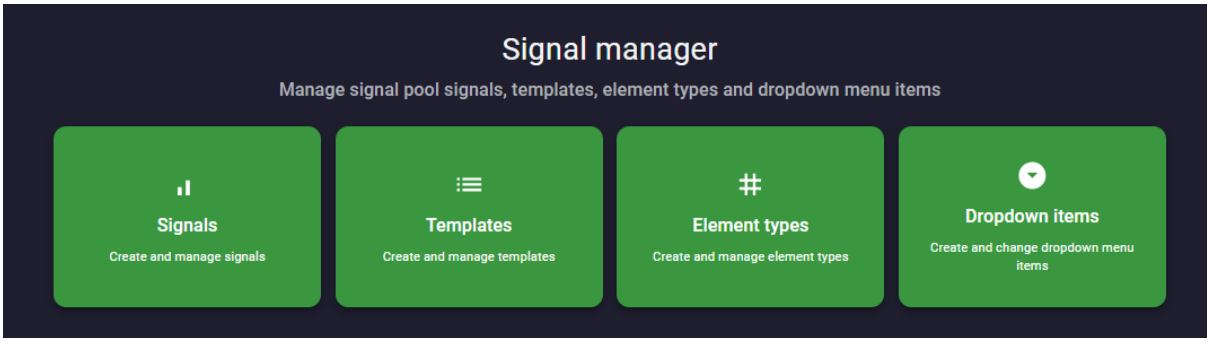


Figure 6.21 – "Signal manager" page for Elvia users. Illustrating the interface when navigating to the "Signal manager" page. Here the Elvia users can navigate to other pages to modify signal pool signals, templates, element types or dropdown menu items.

Figure 6.22 shows the "Dropdown items configurator" page for the Elvia users. This is where they can modify the content that is available in different dropdown menus in the application. The "Active" buttons are used to hide the items from the system dropdown menus without having to delete them. The objective of using dropdown menus in the application is to minimize free-text input, thereby improving standardization and maintainability.

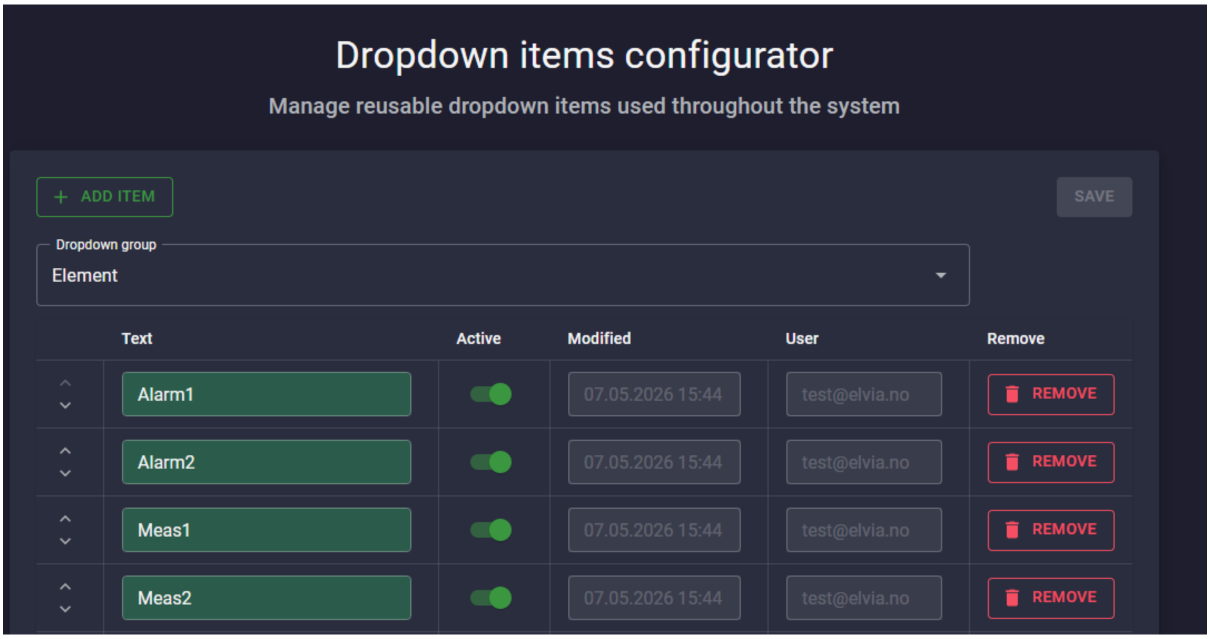


Figure 6.22 – "Dropdown items configurator" page. Illustrating the overview when navigating to the "Dropdown items" page from the "Signal manager" in Figure 6.21. Here Elvia users can modify the content for the different menus that are implemented in the application.

Figure 6.23 shows the “Element type configurator” page for Elvia users. This is where they can modify element types that are used for signal configuration in the application. The figure illustrates that an element type can be added using the “Add element type” button which adds a blank row where the user can specify details for the element type. The configured element types are available for selection in the signal pool when configuring signals.

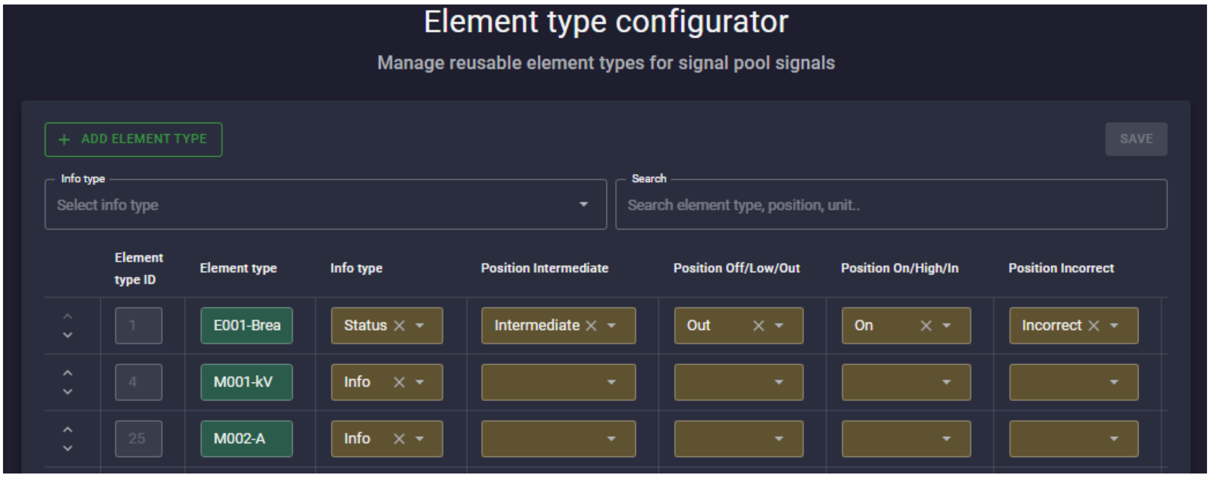


Figure 6.23 – “Element type configurator” page. Illustrating the interface when navigating to the “Element types” page from the “Signal manager” page in Figure 6.21. Here Elvia users can modify element types that are used for signal configuration.

Figure 6.24 shows the “Signal pool configurator” page for the Elvia users. This is where they can modify signals in the signal pool. The objective of the signal pool is to standardize signals that are used in the signal list projects. This is to ensure that different facilities are built as similar as possible. The configured signals are available for selection in the “Template configurator” page or in the engineering section when developing a signal list.

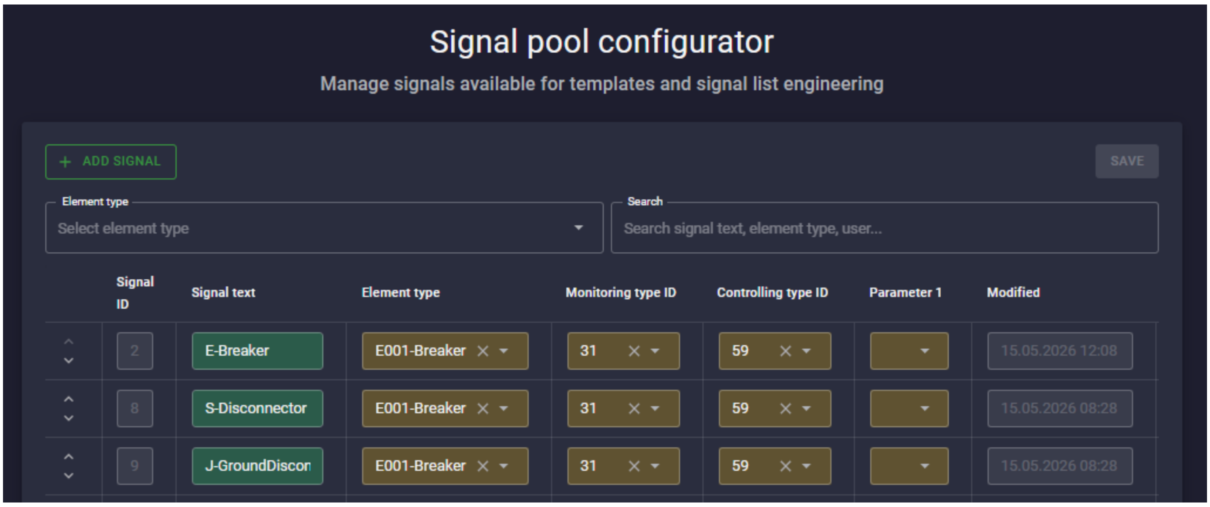


Figure 6.24 – “Signal pool configurator” page. Illustrating the interface when navigating to the “Signals” page from the “Signal manager” page in Figure 6.21. Here Elvia users can modify signal pool signals that are used in template configuration or signal list development.

Figure 6.25 shows the "Template configurator" page for Elvia users. This is where they can modify predefined signal templates. The templates are called assets, and typically refer to a facility component, for example a transformer bay for a substation. The relevant signals for the asset are added from the signal pool and modified according to company standards for SCADA implementation. The template configurator is implemented with functionality for bulk updating the IOA addresses using the "Update IOA addresses" button and functionality to check for duplicate IOA addressing since each signal requires a unique signal. This is the same functionality as described in Section 6.3.3.

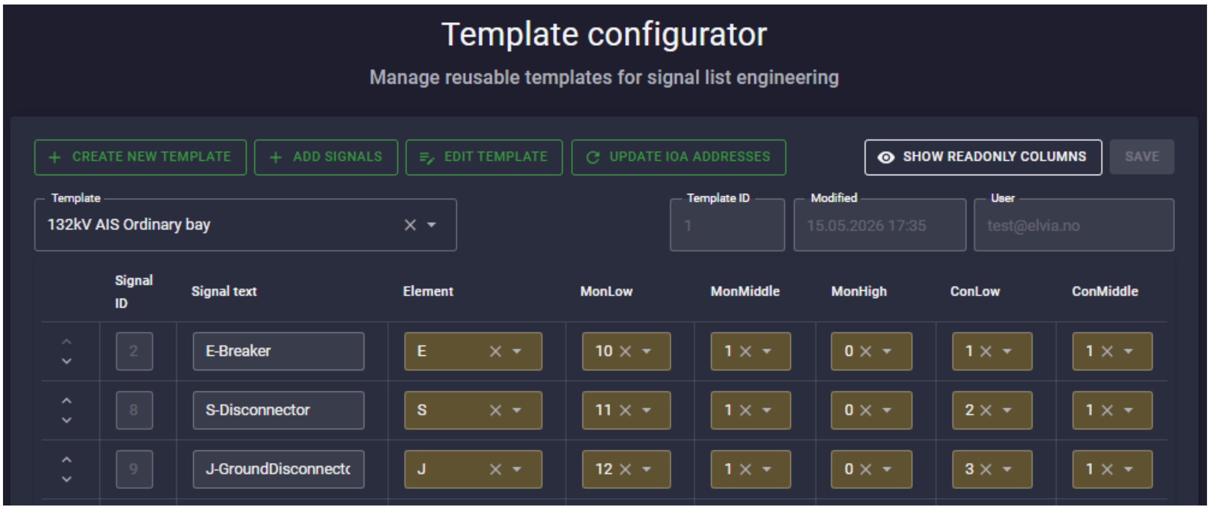


Figure 6.25 – "Template configurator" page. Illustrating the interface when navigating to the "Templates" page from the "Signal manager" page in Figure 6.21. Here Elvia users can modify signal templates. After selecting an existing template or creating a new template the user can add and configure predefined signal from the signal pool as shown in Figure 6.24.

Figure 6.26 shows the signal pool pop-up window for selecting signals into the templates or signal list projects. Here the users can select the relevant signals and insert them into the template asset as shown in Figure 6.25 or signal list asset as shown in Figure 6.10.

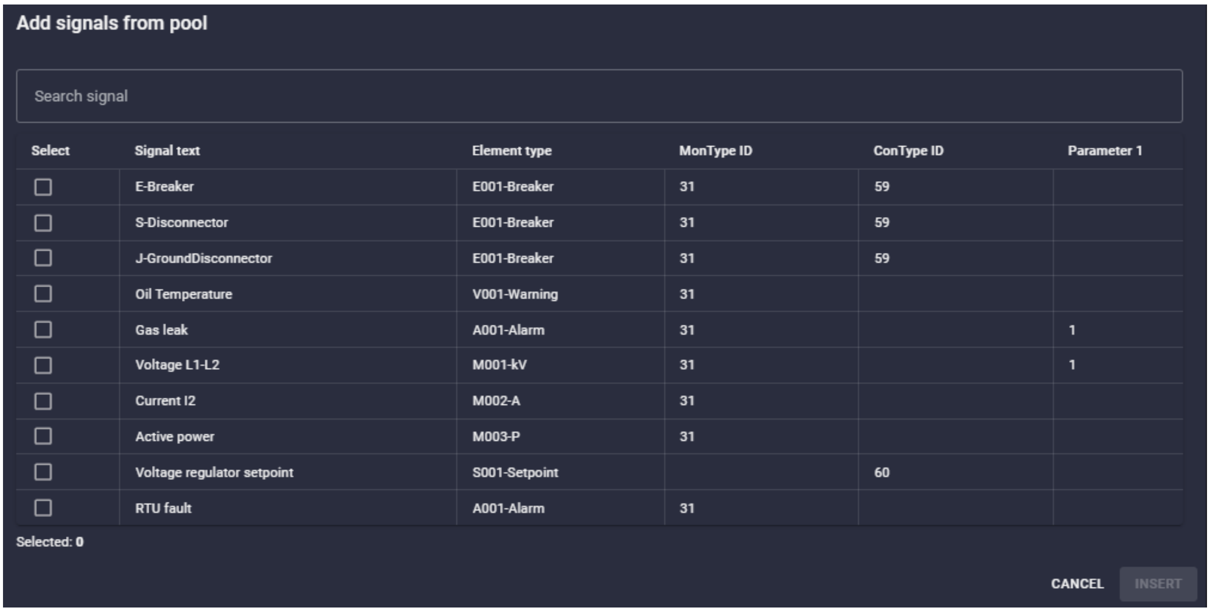


Figure 6.26 – "Signal pool" dialog. Illustrating the pop-up box that opens when pressing the "Add signal" button. Here the user can select which signals to insert to the defined template or signal list.

7 Discussion

This chapter discusses the developed “Signal Management Portal” system and evaluates its performance in relation to the system requirements. The discussion analyses both the strengths and weaknesses of the system and considers its scalability and readiness for deployment for Elvia. Finaly the chapter discusses further development for the system solution.

7.1 Evaluation of the developed system

This section evaluates the developed “Signal Management Portal” system with respect to the defined requirements and system objectives. The section focuses on how the system meets the requirements defined in Chapter 3. The results presented in Chapter 6 illustrate the graphical user interface and demonstrate key system functionalities. These results serve as a basis for the following discussion on how the system meets its core requirements.

The system was deployed to Microsoft Azure, enabling relevant stakeholders to interact and test the application in practice during the project. To minimize the use of available Azure credits, both the database and the application were configured with minimum performance settings. Therefore, the system performance was affected by these limitations and does not reflect the performance of a production ready system setup. However, the Azure deployment made it possible to evaluate the system in a realistic environment and collect feedback from relevant stakeholders. The most significant impact of this configuration was that the database enters auto-pause after 20 minutes of inactivity. As a result of this, the sing-in process could take up to 30 seconds while the database recovers.

The login interface is the first point of system interaction for the users and therefore plays an important role in the overall user experience. The login interface provides a straightforward authentication process, with forgotten password functionality if necessary. It offers a high level of usability, while the access control remains important for basic security purposes suitable for a prototype. However, the system lacks several security features expected from a modern system. Multi-factor authentication, rate limiting, account lockout or authentication logging are missing from the system. These security aspects must be considered before the system can be used for production purposes.

The home page serves as the central interface of the system after sign-in and is designed to present an overview of the main system functionality. The available pages on the home page are restricted based on the user-type. An important requirement was to restrict the “User management” and “Signal administration” pages and functionality to the Elvia users. This ensures that the contractor users cannot access user details or perform modifications to the system configuration. The contractor access rights were limited to the “Creating new project” and “Open existing project” pages only. These restrictions are tested and function as intended in the system.

The page for “Create new project” ensures that key project details are included in the project and serves as an initial framework before adding starting the signal list development. The purpose of the project details is to enable quick navigation between all existing signal list projects and to provide new project stakeholders with a clear understating of project context. These details improved the usability of the system and are useful for project onboarding purposes.

The creation of a new project requires all mandatory fields to be complemented before the project can be created. The dropdown menus are further linked to the SQL database, ensuring consistency and standardization of input data. Additionally, each project is assigned a unique project ID to ensure clear identification.

The “Existing signal list project” page provides the users with an overview of all existing signal list projects and displays key parameters for each project. For the contractors the overview is restricted to only displaying the projects that are assigned with their company, while all projects in the system are visible for Elvia users.

The “Existing signal list project” page has a search and filtering menu that offers the users functionality to find relevant projects. It also has a button for filtering on “My projects” to limit the overview to only show their projects. All these functions support the usability of the system, but usability could be further improved by adding more filtering and sorting functionality.

The “Administrate users” page is limited to the Elvia administrator. It offers functionality to create new users or manage existing user accounts. The system includes several predefined email templates that automatically send emails when users are created, new passwords are generated or existing users are removed, illustrated in Chapter 6. These functions improve the usability and supportability of the system by automating email communication, simplifying user administration, and making it easier to expand the system with additional users.

The “Signal manager” page is limited to Elvia users and provides functionality for managing signals and system configurations that are used throughout the system. An important requirement of the system is to ensure a high level of maintainability. With Elvia’s existing Excel spreadsheet solution for signal list development, even minor modifications to the signal lists must be adjusted repeatedly across multiple entries for all relevant signals. In contrast the “Signal Management Portal” system centralizes this process. Modifications made in the “Signal manager” are automatically applied where relevant throughout the system. The system uses a relational database with stored procedures and primary and foreign keys to ensure that changes are reflected in the relevant tables. This approach improves consistency when updating configurations and improves overall maintainability of the system.

The system is designed so that changes in the “Signal manager” are not automatically applied to existing signal list projects in the system. Instead, modifications are limited to the system configurations, the signal pool and relevant templates. This approach prevents unintended changes to projects that may already be implemented in PAC and SCADA systems, where accurate and consistent documentation is essential. Changing the “source of truth” documentation after implementation could create inconsistencies and have undesirable consequences.

The idea is that if a user discovers an error in the signal configuration during development of a signal list, the error should be corrected in the “Signal manager” rather than locally in the project. This ensures that the correction is applied centralized and is available for future signal list projects. As a result, the correction benefits future users and prevents the same modifications from being corrected the next time.

The “Specific signal list project” page provides different functionalities and views for the engineering and commissioning stages for Elvia and contractors. Both the engineering and commissioning sections are located below the project details section, illustrated in Chapter 6. Each project is assigned a status that determines whether Elvia or the contractors can modify the signal list project during the engineering process. This ensures structured collaboration and a clear overview of the responsibilities throughout the project.

In the current Excel spreadsheet solution, collaboration is managed by exchanging updated versions of the signal list between the parties. This often leads to confusion regarding version control, leading to challenges who have the most updated and correct version and content. In contrast the “Signal Management Portal” system provides a centralized solution where users must update the project status to make modifications, where changing the project status is logged to ensure traceability in the project. This solution ensures that only one version exists at any given time and the latest version will always be available within the system.

The engineering section provides users with the necessary functionality to develop the signal list. During the system development of this section, usability was prioritized by introducing smart functionality intended to improve efficiency when working with signal lists.

The current Excel spreadsheet solution requires a significant amount of manual copy/paste work, which increases the risk of errors and inconsistencies. In contract, the engineering section of the “Signal Management Portal” is designed to minimize free-text input and includes functionality for bulk updating signal list content. This reduces the need for repetitive manual work and helps improve consistency across the signal list.

The solution also includes functionality for hiding signals in assets that are not relevant, helping users maintain a better overview while working with the signal list. In addition, the system includes validation functionality for IOA addresses and prevents duplicate element assignments. In the current Excel solution this typically causes issues during SCADA import.

The commissioning section provides functionality for performing signal testing and documentation for both Elvia and the contractor. This section becomes available when the project is set to “Commissioning” status. The purpose of the commissioning state is to allow both Elvia and contractor engineers to acknowledge and register deviations identified during signal testing on their respective sides. The objective is to achieve zero deviations by ensuring that all signals in the signal list are verified and acknowledged before the commissioning stage is completed. The solution includes logging functionality that records which user acknowledged each signal, that can be useful when reviewing the project at a later stage.

The commissioning section is designed with limited content, focusing on the element necessary for performing signal testing. This helps the users to maintain a clear overview during testing and reducing the risk of confusion, which can be challenging when using the current Excel spreadsheet solution. If errors are discovered during commissioning and signal testing, the project must be returned to engineering status so that the necessary corrections can be made before the signals are tested again in commissioning status. The solution is also implemented according to the requirements with an export to Excel functionality, when paper format is required.

To support users maintaining an overview of the signal testing process, the system provides a dashboard when the project is set to “Finished” status. The view displays a basic overview of the signal testing status and highlights any deviations or signals that have not been acknowledged. In the current Excel solution, users must manually check each row in the Excel list to verify whether the signals have been tested and identify any registered deviations.

In the current Excel spreadsheet solution, signal testing is typically performed by acknowledging or registering deviations in the signal list using text-based comments and color coding. This approach often quickly becomes challenging to manage and lacks clarity. It involves extensive copy/paste operations with a high level of risk for errors and inconsistencies. Since signal testing is often performed under time pressure, a clear and structured solution is important to reduce mistakes.

Based on the overall system evaluation most of the requirements are met. However certain functions require further optimization to fully meet the intended objectives. These are described further in the next section.

7.2 Strengths and limitations of the developed system

This section discusses the strengths and limitations of the developed system, with a focus on non-functional requirements such as usability, robustness, flexibility and scalability. It identifies functionalities that work well, as well as weaknesses and areas that require further development.

7.2.1 Strengths of the developed system

The developed system demonstrates strengths in version control and standardization for signal list development. The solution helps avoid confusion caused by multiple versions of signal list versions being stored in different locations, where it may be unclear which user has the latest version.

The system also focuses on reducing free-text based input as much as possible by providing users with predefined dropdown menus when developing the signal list. This helps to prevent inconsistent data entries. Since the signal list is used as a basis for import to SCADA, standardization and accuracy are important for ensuring a smooth SCADA import process. The system includes mandatory fields to strengthen the level of documentation and reduce the risk of incomplete data.

To support standardization, the system allows Elvia users to create templates using pre-defined signals from the signal pool. This approach makes it easier to develop standardized signal lists for similar facilities for Elvia. Modifications made in the “Signal manager” section are automatically reflected in related parts of the system, ensuring that the signal pool, template and system configuration sections remain updated. This improves the maintainability and usability of the system by reducing the need for repeated manual updates, while also contributing to more standardized signal lists across users and facilities.

The “Signal manager” and engineering section is implemented with functionality intended to increase the efficiency when managing system configurations and developing the signal list. These functionalities are described in Chapter 6 and include functionality for bulk updating of IOA addressing and hierarchical structures, checks for unwanted duplicate entries and asset duplication functions. These were all functions that were requested by Elvia stakeholders.

The system is implemented with change logs to visualize which user has made modifications in the system. The change log also visualizes changes to the project status so users can monitor the project activity. The change log improves the traceability of the system, making it possible to review when and by whom. An improvement to the change log functionality would be to include details about which specific changes were made.

Furthermore, the system is designed with role-based access control to ensure that contractors cannot manage system configurations, and that neither contractor nor regular Elvia users can access user administration. The “Signal manager” section is available to all Elvia users, allowing them to make system modifications during the engineering process of signal list development. The user administration section is restricted to Elvia administrators only. These access rights support controlled data input while still maintaining a flexible system.

The Elvia administrators have through the “User administration” page an overview of registered Elvia and contractor system users. To simplify user administration, the system is implemented with predefined email templates and automatic sending for creating, regenerating passwords and deleting users. The system is configured with a specified email account that sends emails without any need for text modifications by the administrators.

7.2.2 Limitations of the developed system

Despite its strengths, the system does have limitations regarding security, scalability, and usability aspects, that should be included in further development of the system. This section describes these limitations.

As previously mentioned in Section 7.1 the system lacks several security features. Since signal lists are classified as power sensitive documentation the system should include two-factor authentication and functionality for rate limiting, account lockout, authentication logging and password expiration. None of these functions are included in the current system version. The current system only uses a custom username/password authentication solution, where passwords are securely stored as hashed values and verified during login.

Another improvement regarding passwords is to implement a better forgotten password solution. The current version relies on Elvia administrators to manually handle password requests. The current solution introduces unnecessary manual work and could potentially delay user access. The predefined email templates should also be configurable, as these are currently hardcoded in the system, and cannot be modified by users. In future work, these email templates should be made editable for Elvia administrators through the system interface.

There is also system limitations related to multiple users working on the same signal list project while it is in "Engineering" status. Although signal list development typically involves only one Elvia and one contractor engineer, the system may face limitations if multiple users try to edit the same signal list simultaneously. The system does not include functionality for collaborative editing within the same "Engineering" status. As a result, that last user who saves changes may overwrite ongoing changes made by another user editing the project at the same time.

Another improvement would be to introduce more subcategories for organizing data in the system. For example, all alarm element types or all measurement signals could be grouped into categories. In the current version, all rows are listed in a single view, which can quickly become difficult to navigate as the amount of data increases. Categorization could therefore improve the usability by making it easier for users to find the desired data.

The system includes ready-only views for Elvia and contractor users when the project is in relevant "Engineering" statuses. However, users from the same party may still interfere with each other. Future improvements should therefore include collaborative editing functionality or locking mechanism where editing access is restricted to the assigned project engineers.

The "Existing signal list projects" page could be improved by including more filtering and sorting functions. The current solution lists all the projects in one single view and includes some basic filtering functionality. However, additional filtering and sorting options should be implemented to improve the usability. This could include filtering by specific users and sorting projects by creation date in descending order. As the system grows to include hundreds of signal list projects, the current functionality may make it a bit challenging to quickly navigate and identify relevant projects.

The system is designed so that modifications in the "Signal manager" section are reflected to relevant specific system elements. In practice this means that changes in the "Signal manger" section will not trigger changes in any existing signal list projects. The modifications become available if the user manually updates the signal list, for example by selecting a new element from a dropdown menu or inserting the relevant signal again. However, if a user has previously inserted a signal from the signal pool and signal is later assigned a new element type in the signal pool, this will not automatically trigger changes in the existing signal list. In this case the user must delete the relevant signal from the signal list and insert the updated signal from the signal pool again. This is a deliberate choice to avoid unintended changes in existing projects that potentially are already implemented in the PAC system or SCADA. If the signal lists were automatically modified one could potentially lose documentation on how the PAC and SCADA system is configured. However, users may experience this as more restrictive, since the current Excel solution provides more flexibility for making such changes directly in the signal list. The system could therefore be improved by introducing functionality that allows users to choose whether changes in the "Signal manager" should be applied to a specific project if the relevant project is in "Engineering" status.

Another limitation of the system is the lack of SCADA integration. In the current solution, the content in the "Signal manager" must be configured manually. Since much of this data already exists in SCADA, it would be beneficial to implement import functionality from SCADA. For example, element types are already configured in SCADA and could potentially be imported to the "Signal Management Portal" instead of being manually added in the system. Other SCADA integrations should also be explored as part of future work to improve efficiency, data consistency and reduce manual work.

Finally, the system does not include automatic saving functionality. The users are notified by the system if there are unsaved modifications, which help to reduce the risk of users leaving the page without saving their work. However, automatic saving functionality could improve usability and reduce the risk of data loss by ensuring that changes are continuously saved. The system also lacks backup and restore functionality, as well as an undo function. These features would improve robustness and usability by making it possible to recover previous versions or reverse unintended changes.

7.3 Scalability and production readiness of the developed system

The "Signal Management Portal" system has a significant potential for scalability, as the need for such a solution is substantial for Elvia. The current Excel-based solution requires unnecessary maintenance resources, whit a substantial risk of errors.

Signal list development is only one part of the overall process of implementing signals into PAC systems and SCADA. In future development the "Signal Management Portal" could be expanded to support additional tasks related to the SCADA import process. The current Elvia practice involves converting Excel-based signal lists using Python scripts, which generates XML files that can be imported into SCADA to build the hierarchical structuring, signals and addressing. Integrating this functionality into the "Signal Management Portal" could eventually make the system a more complete solution for managing signal lists and preparing them for SCADA implementation.

The current version of the application and SQL database is hosted in Microsoft Azure cloud platform. Since signal lists are classified as power sensitive documentation, a production version of the system should be implemented within existing Elvia infrastructure that is secure and suitable for handling these types of data. The current Azure deployment version should therefore be considered a proof-of-concept solution, chosen primarily to make the system easily accessible for demonstration purposes for relevant stakeholders.

The current system version has not been stress tested with multiple users or large amounts of data. As previously mentioned in Section 7.1 the configuration parameters in Azure are set to a minimum to save credits. Further testing is therefore required to evaluate the system's performance, robustness and scalability under realistic conditions. There are also some question marks whether the architecture is good enough for multiple users, especially multiple users working with the same project in the same status. Therefor it is recommended that experienced Elvia developers become involved in assessing the architecture and design of the system.

The "Signal manager" is configured with one parameter called "Parameter 1". This is just to demonstrate the concept of signal parameterization. In practice a signal can have multiple parameters for different functionalities, but these are kept outside the scope of this thesis. This can be parameters regarding SCADA specific functionalities like flags, priorities, graphical logic etc. Before the system is ready for production use, all necessary parameters must be implemented in the system. It is therefore necessary to involve Elvia stakeholders to identify and implement these parameters.

The solution requires a stronger cyber security focus before it can be used in production. Signal lists are classified as power sensitive documentation, so this is an important requirement before production deployment. For instance, the login section is simplified and should be improved to support modern security requirements. In the current version, it is primarily implemented to demonstrate the concept of authentication and to identify the user type. Elvia network infrastructure resources and developers should therefore be involved in evaluating the overall system and support the necessary improvements before deployment.

7.4 Further development of the system

Further development of the "Signal Management Portal" system should focus on several key areas. The current system should be regarded as a proof-of-concept solution and used as a basis for further development in collaboration with relevant Elvia resources.

The architecture should be implemented and further developed within existing Elvia infrastructure. An important element is the cyber security aspect of the architecture that needs to be thoroughly evaluated. The application should then be stress tested with multiple users working with the system and increasing the amount of data. It is plausible that improvements are necessary in this area since this has not been tested during the project.

The application has significant potential additional functionalities that could simplify the signal list development and SCADA implementation. The application should be expanded with import and export functionalities to SCADA, or possible direct integrations, to ensure that the element types and other relevant configuration data remain updated according to SCADA. Since the signal list forms the basis for building the facility structure in SCADA, functionality that simplifies the process of importing signal list content into SCADA should be further developed.

The application is currently not implemented with automatic saving functionality as mentioned in Section 7.2 and requires manual saving. Modern systems are expected to have this functionality, so this should be implemented to reduce the risk of losing data when using the system.

Finally, as part of the further development process, the application should be tested during an actual Elvia project. Engineers from both Elvia and contractors should try to use the "Signal Management Portal" in collaboration to develop and document a signal list project, and evaluate the usability of the system, identify pain points and suggest improvements.

8 Conclusion

This chapter concludes the project by summarizing the main results of the developed "Signal Management Portal" system. It presents the key findings and reflects on the overall outcome of the developed solution.

8.1 Key findings

- The "Signal Management Portal" system demonstrated a potential to solve many signal management issues for Elvia.
- The system enabled a standardized and structured approach to handling signal lists in collaboration with contractors. It reduces the risk of human errors and manual spreadsheet work. The system reduces maintenance and conflicts and offers version control and access control.
- The system can be used as a basis for further development to support SCADA integration and build a complete system for signal list management and SCADA implementation.
- The system reduces the workflow regarding maintenance and signal list development using built-in functionality that was requested by Elvia stakeholders.
- The SQL database ensured a single source of truth. It maintains one centralized and consistent version of the documentation regarding system configuration and signal list projects.
- Centralized and standardized documentation is crucial for power-sensitive data, as it supports efficient maintenance and future work on the facility.
- The "Signal Management portal" requires further testing and development to ensure sufficient security and robustness under increased user and data load.

8.2 Final remarks

The project resulted in a web-based system for digital collaboration for signal list management. The overall perception is that the system meets the defined requirements for the project. The system has been demonstrated to relevant Elvia stakeholders, and the feedback has been exclusively positive, although there are some question marks regarding the production readiness that is described in Chapter 7.

The most positive feedback regarding the system concerns system maintenance, version control, access control and standardization. The current solution for developing signal lists introduces several challenges related to these areas, and the perception is that the "Signal Management Portal" addresses these issues. It is believed that the system could either be further developed or that its functionality can be implemented into other Elvia systems to simplify the overall signal management process and SCADA implementation.

Topics that are kept outside of the scope must be further evaluated before the system can be used in production. These involve areas like cyber security, network infrastructure, system stress testing and ensuring that the system can handle multiple users working simultaneously.

In conclusion, the thesis demonstrated the feasibility of developing a digital system for collaborative signal list development. The resulting system demonstrated that a centralized solution can improve standardization, traceability, maintainability and usability for signal list development compared to Excel-based solutions.

References

[1] OpenAI, "ChatGPT," OpenAI, 19 01 2026. [Online]. Available: https://chatgpt.com/. [Accessed 19 01 2026].

[2] Wikipedia, "Substation," 07 03 2026. [Online]. Available: https://en.wikipedia.org/wiki/Substation. [Accessed 15 03 2026].

[3] Oslo Byleksikon, "Tonsen transformatorstasjon," 13 03 2026. [Online]. Available: https://oslobyleksikon.no/side/Tonsen_transformatorstasjon. [Accessed 14 03 2026].

[4] International Electrotechnical Commission, "Find out more about IEC 61850," 15 05 2026. [Online]. Available: https://iec61850.dvl.iec.ch/. [Accessed 15 05 2026].

[5] IACS Engineering, "IO List," 12 02 2026. [Online]. Available: https://iacsengineering.com/io-list/. [Accessed 12 02 2026].

[6] Siemens, "Advanced Distribution Management," Siemens, 14 03 2026. [Online]. Available: https://www.siemens.com/en-us/products/gridscale-x/advanced-distribution-management/. [Accessed 14 03 2026].

[7] M. J, "How to read IOA-Adresses from IEC 60870-5-101?," 25 11 2021. [Online]. Available: https://sieportal.siemens.com/en-ww/support/forum/posts/how-to-read-ioa-adresses-from-iec-60870-5-101/272081. [Accessed 24 03 2026].

[8] Norwegian User Group, "Norwegian IEC 60870-5-104 User Conventions," 2023.

[9] Wikipedia, "Modulo," 21 01 2026. [Online]. Available: https://en.wikipedia.org/wiki/Modulo. [Accessed 26 02 2026].

[10] Siemens, "Spectrum Power TM 7 Course Catalogue," Siemens, 2022.

[11] Wikipedia, "Visual Studio," 11 01 2026. [Online]. Available: https://en.wikipedia.org/wiki/Visual_Studio. [Accessed 13 01 2026].

[12] Wikipedia, "Multitier architecture," 21 09 2025. [Online]. Available: https://en.wikipedia.org/wiki/Multitier_architecture. [Accessed 30 01 2026].

[13] M. Chiaramonte, "What is a 3-tier application architecture? Definition and Examples," 13 05 2024. [Online]. Available: https://vfunction.com/blog/3-tier-application/. [Accessed 13 01 2026].

[14] Microsoft, "Overview of ASP.NET Core," 30 01 2026. [Online]. Available: https://learn.microsoft.com/en-us/aspnet/core/overview?view=aspnetcore-10.0. [Accessed 30 01 2026].

[15] H.-P. Halvorsen, "ASP.NET Core Web Programming," 15 04 2026. [Online]. Available: https://halvorsen.blog/documents/programming/web/asp_net.php. [Accessed 15 04 2026].

[16] Microsoft, "ASP.NET Core Blazor," 30 01 2026. [Online]. Available: https://learn.microsoft.com/nb-no/aspnet/core/blazor/?view=aspnetcore-10.0. [Accessed 30 01 2026].

[17] Microsoft, "Razor syntax reference for ASP.NET Core," 30 01 2026. [Online]. Available: https://learn.microsoft.com/nb-no/aspnet/core/mvc/views/razor?view=aspnetcore-10.0. [Accessed 30 01 2026].

[18] Microsoft, "What is ASP.NET Core? | Web Development with .NET for Beginners," 20 02 2024. [Online]. Available: https://www.youtube.com/watch?v=sHDox4Fx6G0&list=PLdo4fOcmZ0oXNZX1Q8rB-5xgTSKR8qA5k&index=1. [Accessed 15 04 2026].

[19] Pragim Technologies, "https://www.youtube.com/watch?v=uuzi3SmCLVo&list=PL6n9fhu94yhVowClAs8-6nYnfsOTma14P," 24 02 2020. [Online]. Available: https://www.youtube.com/watch?v=uuzi3SmCLVo&list=PL6n9fhu94yhVowClAs8-6nYnfsOTma14P. [Accessed 10 03 2026].

[20] Microsoft, "C# for Beginners," 28 04 2026. [Online]. Available: https://learn.microsoft.com/nb-no/shows/csharp-for-beginners/. [Accessed 01 04 2026].

[21] Gardnet AB, "What is MudBlazor?," 30 01 2026. [Online]. Available: https://mudblazor.com/mud/introduction#first-step:-mudblazor-as-a-component-library-. [Accessed 30 01 2026].

[22] ZetBit, "Setup/install MudBlazor in a Blazor Server App," 24 06 2023. [Online]. Available: https://www.youtube.com/watch?v=ztRSbMnhODI&list=PLovgdQDh-lWg1hBcKhLBSK_wRwamMO0Cc. [Accessed 30 03 2026].

[23] T. Litvinavicius, "Icons in MudBlazor and setting it up," 12 03 2021. [Online]. Available: https://www.youtube.com/watch?v=u6b7rCIGZRw&list=PL-TtYNTQK1tPJVzzY8kRWCHrqmK1eiAhu. [Accessed 15 04 2026].

[24] t. Litvinavicius, "MudBlazor Pie Charts in .NET 10 – Interactive Dashboard Example," 30 01 2026. [Online]. Available: https://www.youtube.com/watch?v=FwdWFvbJM3c&list=PL-TtYNTQK1tNh-eMe816TFOh5pN0auS11&index=3. [Accessed 10 04 2026].

[25] Gardnet AB, "Explore MudBlazor," 15 04 2026. [Online]. Available: https://mudblazor.com/docs/overview. [Accessed 15 04 2026].

[26] Microsoft, "Microsoft Azure," 10 03 2026. [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_Azure. [Accessed 15 04 2026].

[27] H.-P. Halvorsen, "Microsoft Azure - SQL Databases and App Services," 04 11 2024. [Online]. Available: https://www.youtube.com/watch?v=ca6Q6LdshIs. [Accessed 01 04 2026].

[28] Microsoft, "What is SQL Server?," 28 01 2026. [Online]. Available: https://learn.microsoft.com/en-us/sql/sql-server/what-is-sql-server?view=sql-server-ver17. [Accessed 02 02 2026].

[29] Wikipedia, "Microsoft SQL Server," 26 11 2025. [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_SQL_Server. [Accessed 13 01 2026].

[30] A. Fernando, "MongoDB:," 17 07 2023. [Online]. Available: https://medium.com/@panaceafernando/mongodb-57ee313cb065. [Accessed 02 02 2026].

[31] EnterpriseDB, "Relational Database vs. Document Database: What’s the Difference?," 08 08 2025. [Online]. Available: https://www.enterprisedb.com/blog/relational-vs-document-database. [Accessed 02 02 2926].

[32] GoWithCode, "SQL vs NoSQL: What’s the Difference?," 24 04 2021. [Online]. Available: https://gowithcode.com/sql-vs-nosql. [Accessed 02 02 2026].

[33] Microsoft, "Tables," 18 11 2025. [Online]. Available: https://learn.microsoft.com/en-us/sql/relational-databases/tables/tables?view=sql-server-ver17. [Accessed 11 03 2026].

[34] Microsoft, "Stored procedures (Database Engine)," 20 11 2025. [Online]. Available: https://learn.microsoft.com/en-us/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver17. [Accessed 11 03 2026].

[35] Microsoft, "Primary and foreign key constraints," 18 11 2025. [Online]. Available: https://learn.microsoft.com/en-us/sql/relational-databases/tables/primary-and-foreign-key-constraints?view=sql-server-ver17. [Accessed 11 03 2026].

[36] GeeksforGeeks, "UML Class Diagram," 13 05 2026. [Online]. Available: https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-class-diagrams/. [Accessed 14 05 2026].

[37] UMLet, "UMLetino," 14 05 2026. [Online]. Available: https://www.umletino.com/umletino.html. [Accessed 14 05 2026].

[38] Microsoft, "Queries," 09 04 2026. [Online]. Available: https://learn.microsoft.com/en-us/sql/t-sql/queries/queries?view=sql-server-ver17. [Accessed 29 04 2026].

[39] H.-P. Halvorsen, "SQL Server," 05 03 2026. [Online]. Available: https://halvorsen.blog/documents/technology/database/sql_server.php. [Accessed 05 03 2026].

[40] T. Saini, "Create, Read Excel File using ClosedXML in C# in ASP.NET CORE," 15 11 2022. [Online]. Available: https://www.youtube.com/watch?v=en0lA0foKYY. [Accessed 15 05 2026].

[41] P. God, "Send Email with a .NET 6 Web API using Mailkit & SMTP," 24 05 2022. [Online]. Available: https://www.youtube.com/watch?v=PvO_1T0FS_A. [Accessed 20 04 2026].

[42] Bitwarden, "What is password hashing?," 12 05 2026. [Online]. Available: https://bitwarden.com/resources/what-is-password-hashing/. [Accessed 12 05 2026].

[43] C. Bernasconi, "How to Hash Passwords with BCrypt in C#," 22 06 2023. [Online]. Available: https://www.youtube.com/watch?v=UNLl4kCpwGo. [Accessed 15 04 2026].

[44] C. McKee, "BCrypt.Net-Next," 09 05 2026. [Online]. Available: https://www.nuget.org/packages/BCrypt.Net-Next?. [Accessed 09 05 2026].

[45] Wikipedia, "WebAssembly," 12 01 2026. [Online]. Available: https://en.wikipedia.org/wiki/WebAssembly. [Accessed 13 01 2026].

[46] P. Rout, "ASP.NET Core Web API Tutorials," 13 01 2026. [Online]. Available: https://dotnettutorials.net/course/asp-net-core-web-api-tutorials/. [Accessed 13 01 2026].

[47] IBM, "What is three-tier architecture?," 13 01 2026. [Online]. Available: https://www.ibm.com/think/topics/three-tier-architecture. [Accessed 13 01 2026].

Appendix A – Project task

FMH606 Master's Thesis

Title: A Digital Collaboration Architecture for Multi-Stakeholder Signal Management in Modern Power Grids

USN supervisor: Hans-Petter Halvorsen, Saba Mylvaganam

External partner: Elvia AS

Task background:

Elvia and the contractor responsible for the Protection Automation and Control System jointly prepare a signal list for the signals that are to be exchanged from the Protection Automation and Control System to Elvia's Network Operational Center and SCADA system.

Currently, collaboration with the contractor regarding these signal lists is challenging due to the lack of suitable collaboration tools and top-down engineering in a System Configuration Tool. At present, there is no dedicated solution for joint work other than Excel documentation, which is uploaded/downloaded by the parties via a shared platform used for dialog, process tracking, documentation and technical clarifications between Elvia and the contractor.

Today's setup can result in the parties working with different versions of the same documents, making it possible to lose track of changes. The objective is to maximize the use of standard templates, typicals, and definitions to streamline engineering and configuration. However, some changes and adaptations will occur during a specific project, and some of these adjustments are typically made continuously during signal testing of the control system, and since this phase is often subject to time pressure, it can be difficult to ensure proper and accurate documentation of all changes from both parties with today's setup.

Another challenge is that signal testing at Elvia can be carried out by multiple individuals within the same project. This can lead to variations in how tests are interpreted and documented in the signal list, especially regarding changes in the signal list.

Task description:

The purpose of this assignment is to develop a **prototype** for a shared digital collaboration framework that supports efficient management of signal lists between Elvia and the engineering contractor. The prototype will not be developed within Elvia's systems but will serve as inspiration for further development. See details below:

- Both parties should have access to a common web-based workspace with appropriate user authentication and role-based access control.
- The system architecture should be evaluated to identify and propose a suitable architectural solution. A three-tier architecture may be considered, but other architectural models are also acceptable.
- The web application should interface with a shared database that stores the signal list and all signals within Elvia's scope. The signal list should be created by selecting signals from a common signal pool, or by using predefined signal list templates defined by Elvia. It should be simple for both parties to add or remove signals, while only Elvia should be permitted to make major modifications to signal properties. The contractor should primarily select signals from the pool or templates and make limited adjustments such as name changes or addressing.
- Revision log should be implemented in the solution.
- The solution should include functionality that allows users to acknowledge and comment during the signal testing process for both parties.
- It should also be possible to export the signal list from the web interface to an Excel file, allowing it to be printed and used on-site during signal testing.
- Part of the assignment will involve defining the user roles and access permissions for the different parties. Another part of the assignment will be to define a more detailed technical specification that describes the requirements of the solution before development. Elvia resources will help to define these.
- Finally, the assignment should evaluate how the proposed solution can interface with the SCADA system enabling a more efficient import process for signal data.

Student category: ITA

Is the task suitable for online students (not present at the campus)? Yes

Practical arrangements:

It is a requirement that the assignment be carried out solely by the online student employed by the company (Elvia AS), as the project work potentially involves access to highly sensitive data. All types of highly sensitive data must be anonymized in the report and presentation.

Supervision:

Generally, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

Signatures:

Supervisor:

Hans-Petter Halvorsen - 2026-02-06

Student:

Knut Erland Strætkvern - 2026-02-06

Appendix B – Project progress plan for "Signal Management Portal" system

Tasks	Ressurs	Process [%]	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15	Week 16	Week 17	Week 18	Week 19	Week 20	Week 21
Teams folder structure/access permissions	Knut Erland	100 %																				
Create Gantt chart	Knut Erland	100 %																				
Create project organization	Knut Erland	100 %																				
Create report document in Teams	Knut Erland	100 %																				
Kick-off meeting with Elvia & USN	Knut Erland	100 %																				
Create report outline	Knut Erland	100 %																				
Finalize project description and title	Knut Erland	100 %																				
Create technical specification	Knut Erland	100 %																				
Create UI sketches of the web interface	Knut Erland	100 %																				
Identify tools and solutions for system development	Knut Erland	100 %																				
Develop UI interface	Knut Erland	100 %																				
Develop back-end logic	Knut Erland	100 %																				
Design/configure Microsoft SQL database	Knut Erland	100 %																				
Report writing	Knut Erland	100 %																				
																						Report
Project status		100 %																				

Coloring:
Ongoing
Completed
Deadline

Appendix C – Dropdown menu configuration

Info type	Element	Time delay	Unit	Monitoring Type ID	Controlling Type ID	Parameter number	Position: Intermediate	Position: Off/Low/Out	Position: On/High/In	Position: Incorrect	Deviation	Modified	User
Info	Alarm1	On	A	31	58	1	Intermediate	Off	On	Incorrect	Fault	02.02.2026 17:50	KESR
Measurement	Alarm2	Off	kV	32	59	2		Low	High		Remaining	05.03.2026 19:50	KESR
Status	Meas1		MW	33	60	3		Normal	Alarm		Discontinued	02.02.2026 17:50	KESR
Command	Meas2		V			4			Triggered			02.02.2026 17:30	KESR
	E		P			5						05.02.2026 19:50	KESR
	Vars1		Q			6						02.02.2026 17:30	KESR
	Vars2					100						05.02.2026 19:50	KESR

Appendix D – Element type configuration

Element ID	Element type	Info type	Signal Type	Position: Intermediate	Position: Off/Low/Out	Position: On/High/In	Position: Incorrect	Unit	Time delay	Modified	User
1	A001	Info	SP		Normal	Alarm				02.02.2026 17:50	KESR
2	U001	Info	SP		Normal	Triggered				05.03.2026 19:50	KESR
3	M001	Measurement	SP					A		02.02.2026 17:50	KESR
4	M002	Measurement	SP					kV		02.02.2026 17:30	KESR
5	E001	Status	DP	Intermediate	Out	In	Incorrect		On	05.02.2026 19:50	KESR

Appendix E – Signal pool configuration

Signal ID	Signal text	Element type	Monitoring Type ID	Controlling Type ID	Parameter 1	Modified	User
1	Gas leak	A001	31			02.02.2026 17:50	KESR
2	Voltage UL1	M001	31			05.03.2026 19:50	KESR
3	E-Breaker	E001	31	59	1	02.02.2026 17:50	KESR

Appendix F – Template configuration

Template	Signal ID	Template ID	Signal text	Element	Element type	MonType ID	MonLow	MonMiddle	MonHigh	MonUnstructured	ConType ID	ConLow	ConMiddle	ConHigh	ConUnstructured	Parameter 1	Modified	User
SSK: 47–132 kV	1	1	Gas leak	Alarm1	A001	30	10	1	132	8651018						1	02.02.2026 17:50	KESR
	2		Voltage UL1	Meas1	M001	30	30	1	132	Calculated						1	02.02.2026 17:10	KESR
	3		Breaker	E	E001	30	50	1	132	Calculated	40	0	0	1	65536	1	02.02.2026 17:20	KESR
Transformer bay: 47–132 kV	1	2	Gas leak	Alarm1	A001	30	10	2	132	Calculated						1	15.03.2026 20:50	GNRI
	2		Voltage UL1	Meas1	M001	30	30	2	132	Calculated						1	05.09.2026 11:50	GNRI
Cable bay: 47–132 kV	3	3	Breaker	E	E001	30	50	3	132	Calculated						1	02.02.2026 17:55	KESR
	1		Gas leak	Alarm1	A001	30	10	3	132	Calculated						1	02.02.2026 14:50	KESR
Line bay: 47–132 kV	2	4	Voltage UL1	Meas1	M001	30	30	4	132	Calculated						1	02.02.2026 17:50	KESR
	3		Breaker	E	E001	30	50	4	132	Calculated						1	02.02.2026 17:52	KESR

Appendix G – Signal list management in engineering mode for Elvia and contractor

Asset	B1	B2	B3	Signal text	Element	Element type	Info type	Text intermediate	Text Off/Low/Out	Text On/High/In	Text fault position	MonType ID	MonLow	MonMiddle	MonHigh	MonUnstructured	ConType ID	ConLow	ConMiddle	ConHigh	ConUnstructured	Parameter 1	Last Modified	User Modified	ACK Elvia	User Elvia	Date Elvia	ACK Contractor	User Contractor	Date Contractor
SSK: 47–132 kV	Holmen-kollen	132	A1	Gas leak	Alarm1	A001	Info		Normal	Alarm		30	10	1	132	8651018						1	02.02.2026 17:50	KESR	☐	KESR	02.02.2026 17:21	☑	GNRI	02.02.2026 16:25
	Holmen-kollen	132	A1	Voltage UL1	Meas1	M001	Measurement					30	30	1	132	Calculated						1	02.02.2026 17:10	KESR	☑	KESR	02.02.2026 17:29	☑	GNRI	02.02.2026 18:23
	Holmen-kollen	132	A1	Breaker	E	E001	Status	Intermediate	Out	In	Fault position	40	50	1	132	Calculated	40	0	0	1	1	1	02.02.2026 17:20	KESR	☐	KESR	02.02.2026 17:26	☐	GNRI	02.02.2026 17:26
	Holmen-kollen	132	T1	Breaker	E	E001	Status	Intermediate	Out	In	Fault position	40	50	2	132	Calculated	40	0	0	2	2	1	05.09.2026 11:50	GNRI	☐	KESR	02.02.2026 17:22	☐	GNRI	02.02.2026 15:20
Cable bay: 47–132 kV	Holmen-kollen	132	K101	Voltage UL1	Meas1	M001	Measurement					20	30	3	132	Calculated						1	02.02.2026 17:55	KESR						
	Holmen-kollen	132	K101	Breaker	E	E001	Status	Intermediate	Out	In	Fault position	40	50	3	132	Calculated						1	02.02.2026 14:50	KESR						
Line bay: 47–132 kV	Holmen-kollen	132	K102	Voltage UL1	Meas1	M001	Measurement					20	30	4	132	Calculated						1	02.02.2026 17:50	KESR						
	Holmen-kollen	132	K102	Breaker	E	E001	Status	Intermediate	Out	In	Fault position	40	50	4	132	Calculated						1	02.02.2026 17:52	KESR						

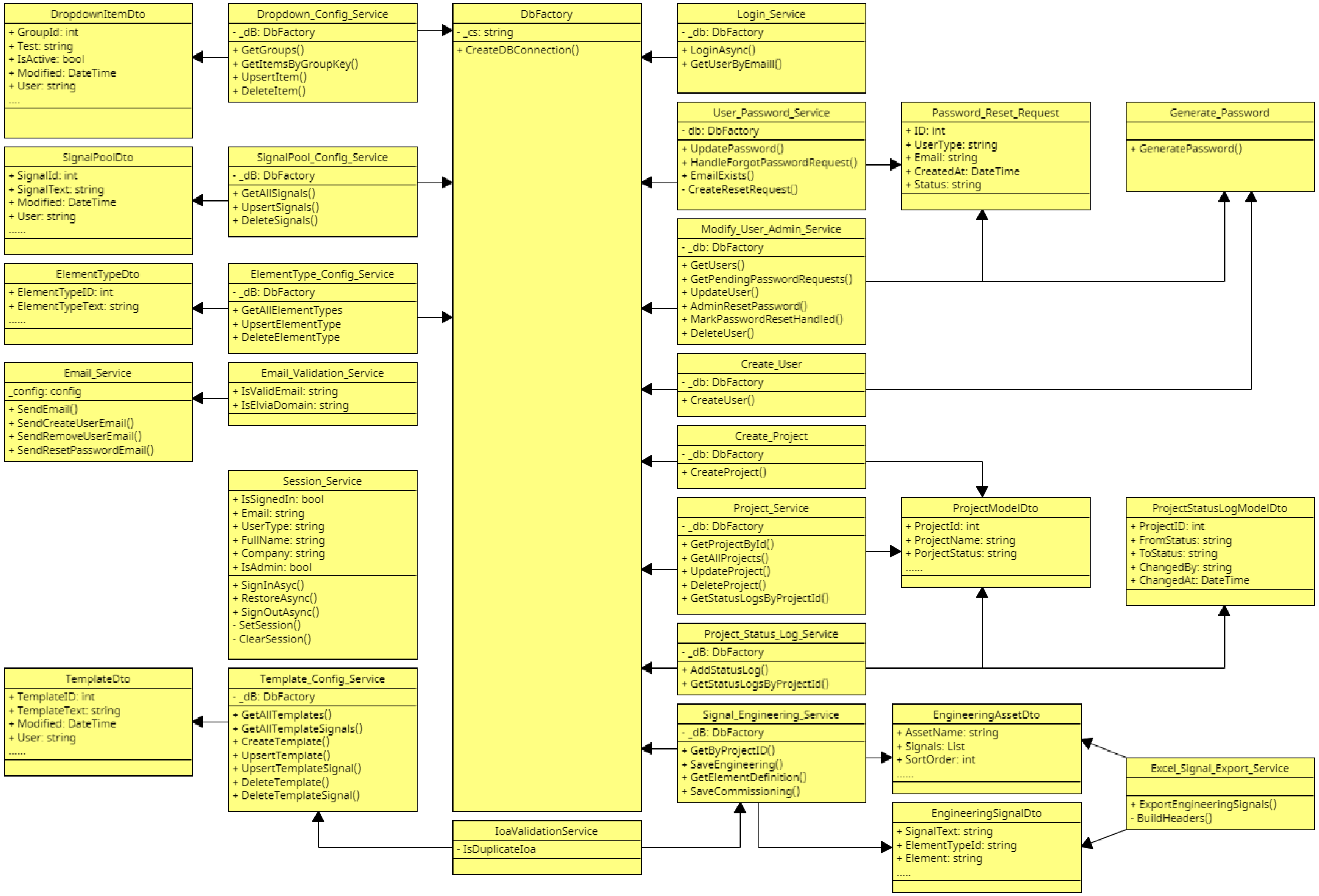
Appendix H – Signal test management in commissioning mode for Elvia

	B1	B2	B3	Signal text	ACK Contractor	ACK Elvia	Deviation Elvia	Elvia Comment	User Elvia ACK	Date Elvia ACK	User Elvia Deviation	Date Elvia Deviation
SSK: 47–132 kV	Holmenkollen	132	A1	Gas leak	☑	☐	Fault (red)	SCADA addressing error			KESR	02.02.2026 16:20
	Holmenkollen	132	A1	Voltage UL1	☑	☑			KESR	02.02.2026 17:20		
	Holmenkollen	132	A1	Breaker	☐	☐	Remaining (yellow)	Pending			KESR	02.02.2026 16:20
Transformer bay: 47–132 kV	Holmenkollen	132	T1	Gas leak	☐	☐	Discontinued (dark gray)	Not in relay plan			KESR	02.02.2026 16:25
	Holmenkollen	132	T1	Breaker		☐						
Cable bay: 47–132 kV	Holmenkollen	132	K101	Voltage UL1		☐						
	Holmenkollen	132	K101	Breaker		☐						
Line bay: 47–132 kV	Holmenkollen	132	K102	Voltage UL1		☐						
	Holmenkollen	132	K102	Breaker		☐						

Appendix I – Signal test management in commissioning mode for contractor

	B1	B2	B3	Signal text	ACK Elvia	ACK Contractor	Deviation Contractor	Contractor Comment	User Contractor ACK	Date Contractor ACK	User Contractor Deviation	Date Contractor Deviation
SSK: 47–132 kV	Holmenkollen	132	A1	Gas leak	☐	☑			GNRI	02.02.2026 16:20		
	Holmenkollen	132	A1	Voltage UL1	☑	☑			GNRI	02.02.2026 17:25		
	Holmenkollen	132	A1	Breaker	☐	☐	Fault (red)	Incorrectly connected in plant			GNRI	02.02.2026 17:25
Transformer bay: 47–132 kV	Holmenkollen	132	T1	Gas leak	☐	☐	Discontinued (dark gray)	Not in relay plan			GNRI	02.02.2026 17:25
	Holmenkollen	132	T1	Breaker								
Cable bay: 47–132 kV	Holmenkollen	132	K101	Voltage UL1								
	Holmenkollen	132	K101	Breaker								
Line bay: 47–132 kV	Holmenkollen	132	K102	Voltage UL1								
	Holmenkollen	132	K102	Breaker								

Appendix J – Class diagram for the developed "Signal Management Portal" system



Appendix K – Visual Studio project structure overview

Folder	Component	Type	Description
/wwwrot/bootstrap	Favicon.png	PNG	Application icon.
/wwwrot/js	Download.js	JavaScript	JavaScript function to download Excel file in browser.
/Components/Layout	LoginLayout.razor	Razor	Application layout for login page.
	MainLayout.razor	Razor	Main layout for application pages.
	NavMenu.razor	Razor	Navigation (hamburger menu) for application pages.
/Components/Pages/1.Login	ForgotPassword.razor	Razor	Page for requesting new password.
	Login.razor	Razor	Page for application sign-in.
/Components/Pages/2.Home	AdministrateUsers.razor	Razor	Page for managing system users.
	CreateNewProject.razor	Razor	Page for creating a new signal list project.
	ExistingProjects.razor	Razor	Page for showing existing signal list projects.
	Home.razor	Razor	Main home page after sign-in.
	SignalManager.razor	Razor	Page for managing signals, templates and system configurations.
/Components/Pages/3.Spesific project	AddAssetDialog.razor	Razor	Dialog for adding assets during engineering.
	EditProjectDialog.razor	Razor	Dialog for editing details of an existing project.
	SelectTemplateDialog.razor	Razor	Dialog for selecting a template.
	SpesificProject.razor	Razor	Main page for a specific signal list project.
	SpesificProjectCommissioning.razor	Razor	Project page for commissioning tasks.
	SpesificProjectEngineeringContractor.razor	Razor	Project page for contractor signal list engineering.
	SpesificProjectEngineeringElvia.razor	Razor	Project page for Elvia signal list engineering.
	UpdateAddressDialog.razor	Razor	Dialog for updating IOA adresses.
/Components/Pages/4.Signal manager	AdministrateDropdownMenu.razor	Razor	Page for managing dropdown menu items.
	AdministrateElementTypes.razor	Razor	Page for managing element types.
	AdministrateSignals.razor	Razor	Page for managing signals.
	AdministrateTemplates.razor	Razor	Page for managing templates.
	SignalPoolPickerDialog.razor	Razor	Dialog for picking signals from the signal pool.
	TemplateEditDialog.razor	Razor	Dialog for editing templates.
/Components/Pages/4.Signal manager	ChangePassword.razor	Razor	Page for changing system user password.
	CreateUserDialog.razor	Razor	Dialog for creating a new system user.
	EditUserDialog.razor	Razor	Dialog for editing an existing system user.
	ModifyUsers.razor	Razor	Page for modifying existing system users.
/Components/Theme	AppTheme.cs	C#	Page for defining application theme, colors and styling.
/Components	_Imports.razor	Razor	Page for common namespaces and imports used by Razor components.

	App.razor	Razor	Page for setting up main structure for rendering the application
	Routes.razor	Razor	Page for handling routing between pages and connecting Razor components.
/appsettings.json	Appsettings.json	Json	File for storing application configuration, like connection string and email account.
/Program.cs	Program.cs	C#	Main startup file.

Appendix L – Stored procedures used in the SQL database

Category	Procedure Name	Description	Related Tables
Create	CreatePasswordResetRequest	Creates a reset password request to Elvia administrator	Passoword_Reset_Request
	CreateProject	Creates a new signal list project	Projects
	CreateUser	Creates a new system user, Elvia or contractor	Users
	DropdownGroupCreate	Creates a new dropdown group (hidden in UI)	Dropdown_Group
	TemplateCreate	Creates a new template	Templates
Get	GetAllProjects	Retrieves all projects	Projects
	GetEngineeringByProjectId	Retrieves engineering section for specific project	Engineering_Assets, Engineering_Signals
	GetPendingPasswordResetRequests	Retrieve reset password requests	Passoword_Reset_Requests
	GetProjectById	Retrieve specific project by id	Projects
	GetProjectStatusLog	Retrieve all project status logs for a project	Project_Status_Log
	GetUsers	Retrieves all system users	Users
	DropdownGroupGetAll	Retrieves all dropdown groups	Dropdown_Group
	DropdownItemGetByGroupKey	Retrieves dropdown items by dropdown group id	Dropdown_Item
	ElementTypeGetAll	Retrieves all element types	Element_Type
	ElementTypeGetByElementType	Retrieves specific element types based on id	Element_Type
	SignalPoolGetAll	Retrieves all signals form signalpool	Signalpool
	TemplateGetAll	Retrieves all signals form templates	Templates
	TemplateSignalGetByTemplateDbId	Retrieves specific template based on id	Templates
	UpdateProject	Updates project details	Projects
	UpdateSignalCommissioning	Updates commissioning details	Engineering_Signals
Delete	DeleteDropdownItem	Deletes dropdown item	Dropdown_Item
	DeleteElementType	Deletes element type	Element_Type
	DeleteEngineeringAsset	Deletes engineering asset in signal list project	Engineering_Asset
	DeleteEngineeringSignal	Deletes engineering signal in signal list project	Engineering_Signal
	DeleteProject	Deletes signal list project. Including details and signal list content.	Projects, Engineering_Asset, Engineering_Signals
	DeleteUser	Deletes system user	Users
	DeleteSignalPoolSignal	Deletes signal from signalpool	Signalpool
	TemplateDelete	Deletes template and assigned signals	Template, Template_Signals
	TemplateSignalDelete	Deletes signals in specific template	Template_Signals
Upsert	DropdownItemUpsert	Inserts/Updates a dropdown item	Dropdown_Item
	ElementTypeUpsert	Inserts/Updates a element type	Element_Type
	SignalPoolUpsert	Inserts/Updates a signal in signalpool	Signalpool
	TemplateSignalUpsert	Inserts/Updates a template signal	Template_Signals
	TemplateUpsert	Inserts/Updates a template	Template
	UpsertEngineeringAsset	Inserts/Updates a engineering asset	Engineering_Asset
	UpsertEngineeringSignal	Inserts/Updates a signal in a engineering asset	Engineering_Asset, Engineering_Signal
	UpsertProjectStatusLog	Updates project status log	Project_Status_Log
	UpdateUser	Updates user account details	Users
	UpdateUserPassword	Updates user account password	Password_Reset_Requests, Users
Other	MarkPasswordResetRequestHandled	Resets password reset request	Password_Reset_Request
	UserEmailExists	Retrieves info if user account already exists when creating new user	Users

Appendix M – Single line diagram and corresponding overview from a signal list project in “Signal Management Portal”