TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Sergei Eensalu 202106IVSM

BUDGET MONITORING WEB APPLICATION FOR MANAGING EU PROJECTS

Master's Thesis

Supervisor: Pelle Jakovits
PhD
Co-supervisor: Gert Kanter
PhD

Tallinn 2025

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Sergei Eensalu 202106IVSM

EELARVE JÄLGIMISE VEEBIRAKENDUS EUROOPA LIIDU PROJEKTIDE HALDAMISEKS

Magistritöö

Juhendaja: Pelle Jakovits
PhD
Kaasjuhendaja: Gert Kanter
PhD

Tallinn 2025

Author’s Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Sergei Eensalu

02.11.2024

Abstract

Master thesis presents the development and implementation of a web-based budget monitoring software tailored for the Tartu City Government. The primary objective is to transition from an Excel-based approach to a robust, scalable, and user-friendly web application. The new system enhances data management, reporting capabilities, and user experience. Key features include real-time project tracking, detailed financial reporting, and role-based access control for employees and project managers. The thesis outlines the requirements analysis, architecture, development tools, and technologies used, and discusses the final outcomes and potential future enhancements.

The thesis is in English and contains 95 pages of text, 8 chapters, 39 figures, 0 tables.

Annotatsioon

Eelarve jälgimise veebirakendus Euroopa Liidu projektide haldamiseks

Käesolev magistritöö käsitleb Tartu linnavalitsusele kohandatud veebipõhise eelarve jälgimise tarkvara väljatöötamist ja rakendamist. Peamine eesmärk on üleminek Excelil põhinevalt lähenemisviisilt tugevale, mastaapselt laienevale ja kasutajasõbralikule veebirakendusele. Uus süsteem täiustab andmehaldus, aruandlus võimalusi ja kasutajakogemust. Olulisteks omadusteks on reaalajas projekti jälgimine, üksikasjalik finantsaruandlus ning rollipõhine juurdepääs töötajatele ja projektijuhtidele. Töö kirjeldab nõuete analüüsi, arhitektuuri, kasutatud arendusvahendeid ja -tehnoloogiaid ning arutleb lõpptulemuste ja võimalike tulevaste täiustuste üle.

Lõputöö on kirjutatud inglise keeles keeles ning sisaldab teksti 95 leheküljel, 8 peatükki, 39 Figuret, 0 tabelit.

List of Abbreviations and Terms

API	Application Programming Interface
AWS	Amazon Web Services
Budget	A financial plan that estimates the revenue and expenses over a specified period for a project or organisation [6]
Completed work / Work history	A record of tasks completed by an employee, including details such as date, duration, and nature of the work performed
Dashboard	An interface that provides a visual overview of key metrics, data, and performance indicators relevant to a specific role or function, facilitating quick decision-making [2]
Employee	An individual who works part-time or full-time under a contract of employment, providing labour to an organisation in return for compensation. In the context of this research, an Employee is a role within the software with specific permissions, allowing access to personal data and limited project information
EMDESK	A project management software solution tailored for Horizon Europe projects, providing tools for planning, controlling, reporting, and collaboration within EU-funded projects [14]
EU	European Union is a political and economic union of 27 member states located primarily in Europe, aiming to ensure the free movement of people, goods, services, and capital within the internal market [12]

Expense	Cost incurred in the process of conducting a project or business activity [4]. These can include salaries, materials, travel, and other operational costs
GDPR	General Data Protection Regulation is a regulation in EU law on data protection and privacy for all individuals within the European Union and the European Economic Area [13]
GCP	Google Cloud Platform is a suite of cloud computing services offered by Google that runs on the same infrastructure that Google uses internally for its end-user products [11]
Labor Cost / Salary	Total expense incurred by an employer to employ workers, including wages, benefits, and taxes [8]. In the context of this research, labour cost includes the employer's expense and taxes, and it is expressed monthly
NPS	Net Promoter Score is a market research metric derived from a single survey question asking respondents to rate how likely they are to recommend a company, product, or service to a friend or colleague [51]. Metric helps measure customer loyalty and predicts business growth by assessing customer satisfaction and likelihood of referral [51]. In the context of this work, author will use this metric to determine "how likely a user is to recommend an application to a colleague."
PM	Person-months is a unit of measurement representing the amount of work performed by an individual in one month [7]. It is used to estimate the effort required to complete a project
Project	A planned, temporary effort aimed at achieving a specific goal [1]. It has a defined beginning and end and can involve creating a unique product, service, or result

Project manager	A professional responsible for planning, executing, and closing projects [5]. Project managers manage teams, resources, and timelines to achieve project objectives [5]. In the context of this research, a Project Manager is a role within the software with broader permissions, including full access to all projects, user data management, and detailed project information
SDK	Software Development Kit is a collection of software tools, libraries, and documentation that developers use to create applications for specific platforms or frameworks [9]
User	An individual who interacts with a system, application, or service. In the context of this research, users can have two roles: Project Manager or Employee, each with different access permissions.
WP	Work package is a detailed, manageable unit of work within a project [3]. It includes deadlines, and resource allocations. In project management, work packages help break down complex projects into smaller, more manageable components [3].

Table of Contents

List of Figures

1 Introduction

Effective budget monitoring is a critical function for governmental organisations to ensure financial accountability and resource optimization [15]. However, traditional budget monitoring tools often fall short in meeting the dynamic and complex needs of modern governance, particularly in a city government setting like Tartu. This thesis addresses the pressing need for a more robust, scalable, and user-friendly budget monitoring system tailored to the specific requirements of the Tartu City Government.

The current landscape of budget monitoring in Tartu relies heavily on Excel-based solutions, which, while flexible, are prone to issues such as data integrity, scalability, and limited collaborative capabilities. These limitations impede the city's ability to efficiently manage budgets, track expenses, and ensure transparency across various departments and projects. Consequently, there is a compelling need to transition to a more advanced and integrated software solution that can handle these complexities with greater ease and reliability.

This thesis aims to develop and implement a web-based budget monitoring software that leverages modern technologies to overcome the limitations of traditional tools. The primary objective is to create a system that not only enhances data management and reporting capabilities but also improves user experience through a streamlined and intuitive interface.

1.1 Research Problem

The main research problem addressed in this thesis is the development of a comprehensive budget monitoring software that meets the evolving needs of the Tartu City Government. This involves transitioning from a fragmented, Excel-based approach to an integrated web-based solution that ensures scalability, accuracy, and ease of use. The research questions guiding this thesis include:

- How can budget monitoring be optimised using modern web technologies?
- What are the key functional and nonfunctional requirements for a robust budget monitoring system in a city government context?
- How can the new system improve data integrity, scalability, and user collaboration compared to traditional methods?

- How can the data entry process be simplified to enhance user experience and minimise errors?

1.2 Problem Solving Approach

To address the research questions, the thesis will follow a structured methodology comprising several key steps:

1. **Requirements Analysis:** Conducting comprehensive stakeholder interviews to gather detailed functional and non-functional requirements specific to the Tartu City Government's needs.
2. **System Design:** Leveraging modern web technologies and development frameworks to design a scalable, user-friendly web application. This phase includes creating detailed designs for both the database architecture and the frontend user interface.
3. **Development:** Building the application based on the system design. This involves developing the frontend to ensure an intuitive user experience and integrating the database to manage business logic and data processing.
4. **Implementation and Testing:** Deploying the system and conducting thorough testing to ensure all components function correctly. This includes user acceptance testing (UAT) to validate the system's usability and reliability. Iterative feedback from potential end-users will be incorporated to refine the application.
5. **Evaluation:** Assessing the system's performance using user feedback and predefined metrics to identify areas for improvement. This step will help validate the effectiveness of the solution in meeting the outlined objectives.
6. **Documentation:** Documentation will include the master thesis, documentation of accounts and passwords, access instructions for different environments, and recommendations for future development options to support ongoing use and future development of the software.

1.3 Scope

The scope of this thesis encompasses the development of an initial version of a web-based budget monitoring software tailored to the needs of the Tartu City Government. This initial version aims to replace the existing Excel-based solutions with a more integrated and efficient system. The focus will be on essential features such as project and financial management, real-time tracking of work hours and expenses, and user-friendly reporting functionalities. The goal is to create a foundation that addresses the immediate needs of the Tartu City Government while providing a scalable platform for future enhancements and additional features.

1.4 Thesis Outline

This thesis is structured to systematically address the development and implementation of a new budget monitoring software tailored for the Tartu City Government. The following sections provide a detailed roadmap of the research, design, and evaluation process undertaken in this project:

Section 2 - Background - overview of the current state of budget monitoring in governmental organisations, emphasising the limitations of existing tools and the need for a more advanced solution tailored to the Tartu City Government's requirements. This section also includes an analysis of existing budget monitoring solutions, including Excel-based approaches and web-based software, discussing their advantages and limitations.

Section 3 - Requirements Analysis - detailed identification of the functional and non-functional requirements of the new budget monitoring software, including user roles, permissions, and specific needs of the Tartu City Government.

Section 4 - Development - development process of the new budget monitoring software. This section covers the selection of development tools and technologies, the implementation process, and the application and database architecture design. It also addresses GDPR compliance and business logic.

Section 5 - Final Outcome - final outcome of the project, describing the different user roles and their respective permissions, and the main features available on the user and project dashboards for employees and project managers.

Section 6 - User Research - comprehensive user research methodology and result, focusing on how the software was tested and validated with real users to ensure it meets their needs. This section also includes feedback collection to refine the system.

Section 7 - Possibilities for Further Development - potential future enhancements, such as migrating to a relational SQL database, developing a robust backend, improving reporting features, incorporating user feedback, and upgrading the budget calculation logic to include Estonian holidays.

2 Background

The landscape of budget monitoring has evolved significantly over the past few decades, influenced by advancements in technology and the increasing complexity of financial management in governmental organisations [16]. Weikart, Chen, and Sermier have written [17] that traditionally, tools like Excel have been the mainstay for budget tracking due to their flexibility and widespread familiarity. Despite their advantages, these tools come with inherent limitations, especially in handling large datasets, ensuring data accuracy, and facilitating collaborative efforts.

Weikart, Chen, and Sermier have written [17] that Excel-based solutions, while versatile, are prone to manual errors, lack real-time data synchronisation, and often suffer from scalability issues. One of the reasons highlighted was that as datasets grow and the number of users increases, these spreadsheets can become cumbersome, leading to inefficiencies and potential data integrity problems. Weikart, Chen, and Sermier also highlighted that the collaborative features of Excel are limited, often resulting in version control challenges when multiple users are involved in budget monitoring tasks.

Weikart, Chen, and Sermier also highlighted in their study [17] that the need for more sophisticated budget monitoring systems has become apparent, particularly in governmental settings where financial accountability and resource optimization are paramount. The Tartu City Government, like many other municipal bodies, has faced challenges with its traditional Excel-based approach. The system's limitations in handling complex financial data and providing a seamless user experience have highlighted the need for a more advanced solution.

The emergence of web-based applications offers a promising alternative. These applications provide centralised data management, enhanced security measures, and better support for collaborative work environments. Web-based systems can integrate various functionalities, streamline processes, and offer more robust data analytics capabilities, making them well-suited for the dynamic needs of modern governance.

Implementing a web-based budget monitoring system for the Tartu City Government requires a thorough understanding of both the technological landscape and the specific needs of the city's administrative framework. By leveraging modern web technologies, such a system can address

the shortcomings of traditional tools, offering improved data integrity, scalability, and user collaboration.

The shift to a web-based solution is not just about adopting new technology; it also involves rethinking the entire approach to budget monitoring. This includes developing new methodologies for data entry, ensuring compliance with regulations like the General Data Protection Regulation (GDPR), and creating intuitive user interfaces that simplify complex tasks. The goal is to create a system that is not only more efficient but also easier to use, thereby enhancing the overall effectiveness of budget monitoring within the Tartu City Government.

This thesis explores these themes, presenting a comprehensive solution that integrates the latest technological advancements with practical applications tailored to the needs of the Tartu City Government. The following sections will delve into the specifics of the proposed system, examining its development, implementation, and potential for future enhancements.

2.1 Existing Solutions

Effective budget monitoring is essential for ensuring that financial resources are managed and utilised efficiently [15]. Various solutions are available for this purpose, ranging from simple spreadsheet tools like Excel to comprehensive web-based applications developed by the European Commission.

2.1.1 Excel-Based Approach

Benninga, S. explains [19] that Excel is a widely-used tool for budget monitoring due to its flexibility and accessibility. It allows users to create custom templates for tracking expenses, income, and various financial metrics. Using Excel, budget managers can set up spreadsheets that include formulas for automatic calculations, pivot tables for data analysis, and visualisations such as charts and graphs for easier interpretation of financial data.

Pros [19]:

- **Customization**: Users can tailor spreadsheets to their specific needs.
- **Cost-Effective**: Excel is often included in existing office software packages.
- **Familiarity**: Users are already familiar with Excel, reducing the learning curve.

Cons [18, 50]:

- **Scalability**: Handling extensive datasets can become unwieldy.
- **Collaboration**: Concurrent multi-user access is limited and can lead to version control issues.
- **Error-Prone**: Manual data entry increases the risk of errors.

Below are examples of current Excel-based solutions, including those used by the client, detailing project timesheets, employee timesheets, and workload and budget allocations.

2.1.1.1 Projects Timesheet

Figure 1 illustrates an example of projects that are split into work packages, providing detailed information for each work package, including start date, end date, and person-months allocated. The data is organised in a tabular format, allowing for easy tracking and management of project timelines and resource allocation. Multiple projects and their respective work packages are included, with corresponding months and work durations clearly outlined.

Projekt 1	Pakett	WP1	WP2	WP3	WP4	WP5	WP6
	Töökuud	1	1	1	4	1	1
	Algus	M1 09.2021	M1 09.2021	M1 09.2021	M1 09.2021	M8 04.2022	M1 09.2021
	Lõpp	M36 08.2024	M10 06.2022	M16 12.2022	M36 08.2024	M36 08.2024	M36 08.2024
	Kestus kuud	36	10	16	36	29	36

Projekt 2	Pakett	WP1	WP2	WP3	WP4	WP5	WP6	WP7
	Töökuud	3	19.2	1	3	1	10	1
	Algus	M1 03.2021	M12 09.2021	M13 03.2022	M3 05.2021	M1 03.2021	M1 03.2021	M1 03.2021
	Lõpp	M40 06.2024	M42 08.2024	M48 12.2024	M48 02.2025	M48 02.2025	M48 02.2025	M48 02.2025
	Kestus kuud	40	31	36	46	48	48	48

Projekt 3	Pakett	WP1	WP2	WP3	WP4	WP5	WP6	WP7	WP8	WP9
	Töökuud	0	7	0	0	8.5	23	1	1	0.5
	Algus	M1 06.2022	M1 06.2022	M1 06.2022	M1 06.2022	M1 06.2022	M1 06.2022	M1 06.2022	M1 06.2022	M1 06.2022
	Lõpp	M24 05.2024	M48 05.2026	M48 05.2026	M40 05.2025	M48 05.2026	M48 05.2026	M48 05.2026	M48 05.2026	M48 05.2026
	Kestus kuud	24	48	48	41	48	48	48	48	48

Projekt 4	1/11/2021	31.12.2023

Projekt 5	Pakett	WP1	WP2	WP3	WP4	WP5
	Töökuud	1	2	1	3	1
	Algus	M1 10.2022	M1 10.2022	M3 12.2022	M14 11.2023	M1 10.2022
	Lõpp	M24 09.2024	M10 07.2023	M14 11.2023	M23 08.2024	M24 09.2024
	Kestus kuud	24	10	12	10	24

	Jan-21	Feb-21	Mar-21	Apr-21	May-21	Jun-21	Jul-21	Aug-21	**Sep-21**	**Oct-21**	**Nov-21**
Projekt 1									M1	M2	M3
WP1									0.03	0.03	0.03
WP2									0.10	0.10	0.10
WP3									0.06	0.06	0.06
WP4									0.11	0.11	0.11
WP5											
WP6									0.03	0.03	0.03
Kokku									**0.33**	**0.33**	**0.33**

	Jan-21	Feb-21	**Mar-21**	**Apr-21**	**May-21**	**Jun-21**	**Jul-21**	**Aug-21**	**Sep-21**	**Oct-21**	**Nov-21**
Projekt 2			M1	M2	M3	M4	M5	M6	M7	M8	M9
WP1			0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
WP2											
WP3											
WP4					0.07	0.07	0.07	0.07	0.07	0.07	0.07
WP5			0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
WP6			0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
WP7			0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Kokku			**0.33**	**0.33**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**	**0.39**

Figure 1. Projects timesheet.

2.1.1.2 Employee Timesheet

The image shows an employee timesheet for October 2020 from the University of Tartu. Each row represents a workday where the employee logs the date, job description, and hours spent on different projects. The timesheet includes a summary of total hours and percentages of work distribution across projects. It also has sections for participant and supervisor signatures to confirm the accuracy of the reported data. Figure 2 illustrates an example of an employee

timesheet, detailing daily tasks, hours spent, total hours, and work distribution.

Name of the participant: **Nimi Perekonnanimi**		Name of the	**RADON**	
		Project code:	**825040**	
Date	**Job description**	**RADON**	**Other**	**SUM**
10/1/2020	RADON-SODALITE workshop.	2	6	**0**
10/2/2020		0	8	**8**
10/3/2020		0	0	**0**
10/4/2020		0	0	**0**
10/5/2020	PMB call. Project management.	1	7	**8**
10/6/2020		0	8	**8**
10/7/2020	WP2 call.	1	7	**8**
10/8/2020	Project management.	1	7	**8**
10/9/2020		0	8	**8**
10/10/2020		0	0	**0**
10/11/2020		0	0	**0**
10/12/2020		0	8	**8**
10/13/2020		0	8	**8**
10/14/2020	Project management. Providing PM estimation	2	6	**8**
10/15/2020	WP3, WP5, WP6 calls.	2	6	**8**
10/16/2020		0	8	**8**
10/17/2020		0	0	**0**
10/18/2020		0	0	**0**
10/19/2020	PMB call. Project management.	1	7	**8**
10/20/2020	WP7 call.	1	7	**8**
10/21/2020	WP2 call.	1	7	**8**
10/22/2020		0	8	**8**
10/23/2020		0	8	**8**
10/24/2020		0	0	**0**
10/25/2020		0	0	**0**
10/26/2020	Project management.	2	6	**8**
10/27/2020	PMB call. ATC call. Project management.	2	6	**8**
10/28/2020		0	8	**8**
10/29/2020	WP3 call.	1	7	**8**
10/30/2020		0	8	**8**
10/31/2020		0	0	**0**
Workhours summary		**17**	**159**	**176**
Percent		**9.66%**	**90.34%**	**100.00%**
Participant				
Confirm that data is correct				
Signature:			Date:	10/30/2020
First and last	Pelle Jakovits			
Person in chief inthe project (RADON)				

Figure 2. Employee Timesheet.

2.1.1.3 Employee Workload and Budget Allocation

Figure 3 illustrates an example of detailed breakdown of each employee's workload across various projects. It lists their monthly salary, percentage of time allocated to each project, and corresponding financial contributions. Columns for different projects indicate employee participation, salary distribution, taxes, and total costs, allowing for clear tracking of employee involvement and budget allocation per project.

2022		Jaanuar	Veebruar	Märts	Aprill	Mai	Juuni	Juuli	August	September	Oktoober
EnlightenMe	123,964.44 €	1,802.96 €	1,287.83 €	1,833.06 €	1,833.06 €	1,647.56 €	1,799.61 €	1,256.45 €	1,876.67 €	2,302.24 €	2,060.52 €
Microgrids	59,902.06 €	1,394.87 €	1,655.78 €	1,525.32 €	1,525.32 €	1,525.32 €	1,527.65 €	1,536.54 €	1,917.57 €	2,277.31 €	2,274.60 €
2ISECAP	31,500.00 €	481.68 €	280.98 €	254.22 €	254.22 €	254.22 €	254.22 €	133.16 €	1,370.58 €	2,283.21 €	2,421.78 €
TREASoURcE	**164,687.57**								1,968.59 €	2,986.42 €	3,197.82 €
oPEN LAB									198.95 €	- €	- €
BUILD	**27,740**										120.42 €

Jaanuar		EnlightenME					Microgrids				
	Töötasufond kuus	Osalus	Maksud	Brutotasu	Töötasufond	**Makstud**	Osalus	Maksud	Brutotasu	Töötasufond	**Makstud**
Töötaja 1	3478.8						30%	263.62 €	780.02 €	1,043.64 €	**1,043.64 €**
Töötaja 2	2408.4	50%	304.18 €	900.02 €	1,204.20 €	**1,204.20 €**					
Töötaja 3	2676.0	10%	67.60 €	200.00 €	267.60 €	**267.60 €**					
Töötaja 4	2341.5	5%	29.57 €	87.50 €	117.08 €	**117.08 €**	15%	88.72 €	262.51 €	351.23 €	**351.23 €**
Töötaja 5	2140.8	10%	54.08 €	160.00 €	214.08 €	**214.08 €**					
	Kokku:				1,802.96 €	**1,802.96 €**				1,394.87 €	**1,394.87 €**

Veebruar		EnlightenME					Microgrids				
	Töötasufond kuus	Osalus	Maksud	Brutotasu	Töötasufond	**Makstud**	Osalus	Maksud	Brutotasu	Töötasufond	**Makstud**
Töötaja 1	4014						30%	304.18 €	900.02 €	1,204.20 €	**1,204.20 €**
Töötaja 2	2676	20%	135.19 €	400.01 €	535.20 €	**575.34 €**					
Töötaja 3	3211.2	10%	81.11 €	240.01 €	321.12 €	**321.12 €**					
Töötaja 4	3010.5	5%	38.02 €	112.50 €	150.53 €	**150.53 €**	15%	114.07 €	337.51 €	451.58 €	**451.58 €**
Töötaja 5	2408.4	10%	60.84 €	180.00 €	240.84 €	**240.84 €**					
	Kokku:				1,247.69 €	**1,287.83 €**				1,655.78 €	**1,655.78 €**

Märts		EnlightenME					Microgrids				
	Töötasufond kuus	Osalus	Maksud	Brutotasu	Töötasufond	**Makstud**	Osalus	Maksud	Brutotasu	Töötasufond	**Makstud**
Töötaja 1	3746.4						30%	283.90 €	840.02 €	1,123.92 €	**1,123.92 €**
Töötaja 2	2542.2	20%	128.43 €	380.01 €	508.44 €	**508.44 €**					
Töötaja 3	2943.6	10%	74.36 €	220.00 €	294.36 €	**294.36 €**					
Töötaja 4	2676	5%	33.80 €	100.00 €	133.80 €	**133.80 €**	15%	101.39 €	300.01 €	401.40 €	**401.40 €**
Töötaja 5	2274.6	10%	57.46 €	170.00 €	227.46 €	**227.46 €**					
Töötaja 6	669.0	100%	168.99 €	500.01 €	669.00 €	**669.00 €**					
	Kokku:				1,833.06 €	**1,833.06 €**				1,525.32 €	**1,525.32 €**

Figure 3. Employee workload and budget allocation.

The Excel-based approach, while flexible and cost-effective, is not ideal for the client due to its limitations in collaboration and manual data entry. Handling extensive datasets can become cumbersome, and the limited concurrent multi-user access leads to version control issues. Additionally, manual data entry increases the risk of errors, necessitating dedicated employees solely for entering data into Excel, which is inefficient and prone to mistakes. Consequently, this

solution does not meet the client's needs for efficient and accurate budget monitoring with different teams.

2.1.2 Web-Based Software

The official EMDESK website states [14] that their web-based software offers a comprehensive solution tailored for Horizon Europe projects, streamlining EU project management in a collaborative, efficient, and GDPR-compliant manner. This all-in-one software enables users to handle project and financial management, planning, control, reporting, and collaboration, keeping everything centralised and always current. Key features include real-time visibility into project progress, finances, and deadlines; time and expense tracking for simplified financial reporting; secure document and report sharing; and easy generation of compliant proposals and reports. Additionally, EMDESK enhances communication with features such as messaging, reminders, discussions, email distribution lists, and video conferencing, effectively reducing the need for back-and-forth emails. The software is customizable for other EU and nationally funded projects.

Pros [14]:

- **Comprehensive Features**: Includes project and financial management, planning, reporting, and collaboration tools.
- **Real-Time Updates**: Maintains up-to-date visibility of project progress, finances, and deadlines.
- **Enhanced Communication**: Enhances communication with tools for messaging, reminders, discussions, email distribution lists, and video conferencing.
- **Secure**: Ensures data security and compliance with GDPR standards.
- **Customizable**: Can be tailored for various EU and nationally funded projects.

Cons:

- **Complexity**: The extensive features might be overwhelming for users with simpler needs.
- **Cost**: The software is relatively expensive, with plans starting at €15.00 per month per user and €134.00 per month for single project teams [14].
- **Overly Complex for Basic Needs**: Clients who only need basic time tracking and reporting might find EMDESK's functionality excessive.

Below are examples of the EMDESK web-based software solution, including its work plan and project analytics sections, as well as reasons why it may not be suitable for the client's needs.

2.1.2.1 Work Plan Overview

The EMDESK web solution interface displays the work plan section with a table view of various work packages (WP) within a project. Each row includes detailed information such as the work package name, assignees, lead participant, start and end dates, actual start and end dates, budget, expenses, and the current status. The interface also offers options for exporting data, adding filters, and modifying table settings. Figure 4 illustrates an example of the EMDESK work plan section, detailing various work packages, assignees, and status.

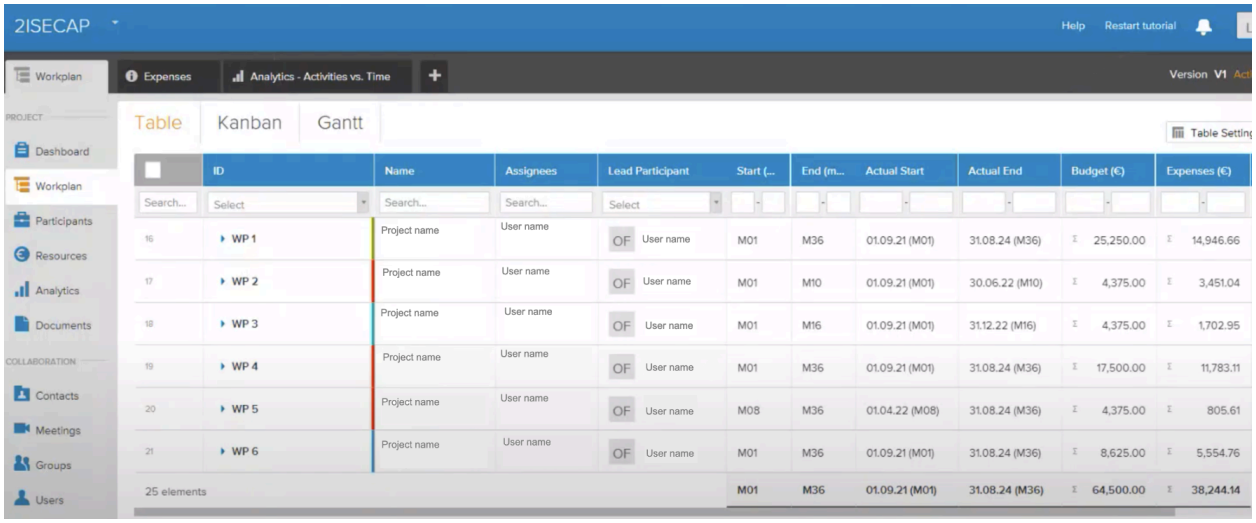
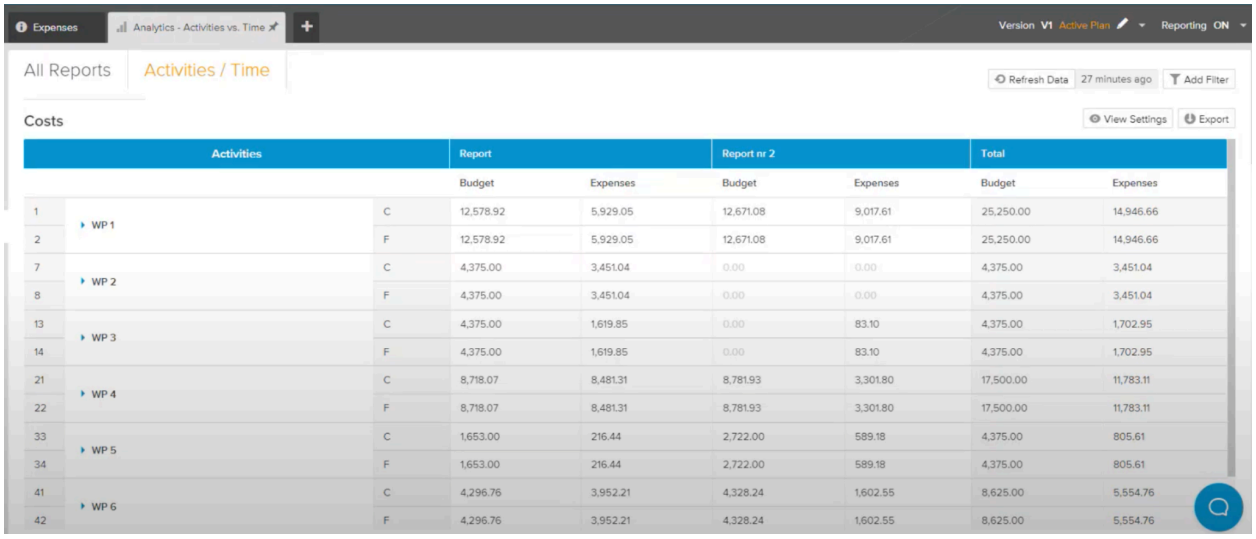


Figure 4. Work plan overview.

2.1.2.2 Project Analytics

The EMDESK web solution interface shows the Analytics section with a focus on activities and time. The table displays costs for various work packages (WPs), listing the budget and expenses for different reports and the total budget and expenses. Each row corresponds to a work package, providing a detailed view of financial data. The interface includes options to refresh data, add filters, adjust view settings, and export data. Figure 5 illustrates an example of the EMDESK analytics section, detailing costs for various work packages, including budget and expenses.



	Activities		Report		Report nr 2		Total	
			Budget	Expenses	Budget	Expenses	Budget	Expenses
1	WP 1	C	12,578.92	5,929.05	12,671.08	9,017.61	25,250.00	14,946.66
2		F	12,578.92	5,929.05	12,671.08	9,017.61	25,250.00	14,946.66
7	WP 2	C	4,375.00	3,451.04	0.00	0.00	4,375.00	3,451.04
8		F	4,375.00	3,451.04	0.00	0.00	4,375.00	3,451.04
13	WP 3	C	4,375.00	1,619.85	0.00	83.10	4,375.00	1,702.95
14		F	4,375.00	1,619.85	0.00	83.10	4,375.00	1,702.95
21	WP 4	C	8,718.07	8,481.31	8,781.93	3,301.80	17,500.00	11,783.11
22		F	8,718.07	8,481.31	8,781.93	3,301.80	17,500.00	11,783.11
33	WP 5	C	1,653.00	216.44	2,722.00	589.18	4,375.00	805.61
34		F	1,653.00	216.44	2,722.00	589.18	4,375.00	805.61
41	WP 6	C	4,296.76	3,952.21	4,328.24	1,602.55	8,625.00	5,554.76
42		F	4,296.76	3,952.21	4,328.24	1,602.55	8,625.00	5,554.76

Figure 5. Project analytics.

2.1.2.3 Suitability for Tartu City Government

Despite its comprehensive features, EMDESK may not be suitable for the Tartu City Government due to several factors:

1. **Cost:** The pricing plans, starting at €15.00 per month per user for the Team plan and €134.00 per month for the Project plan, can be prohibitive for a municipal government with limited budgets or smaller projects. This cost is particularly significant when managing multiple projects simultaneously.
2. **Complexity:** The extensive features of EMDESK, while beneficial for large-scale EU projects, might be excessive for the Tartu City Government, which primarily needs to track project budgets and work hours. The city's basic time tracking, reporting, and budget management requirements can be fulfilled with more straightforward and cost-effective solutions.
3. **Specific Requirements:** The Tartu City Government needs a tool for simultaneously managing multiple projects, user identification (registered and logged-in users), and detailed tracking of project budgets and work hours. Each project requires two main sections: budget and work time tracking. Additionally, the system needs to handle at least five types of expenses (labour, travel, other costs, equipment/investments, and overhead) and provide precise entry of expenses by date.

4. **Real-Time Adjustments:** The city's requirement includes real-time adjustments of the project timeline and budget usage based on actual data entries. The system should clearly distinguish whether the financials and time usage are on schedule or if there are significant deviations. This dynamic adjustment capability is crucial for effective project monitoring.

Considering these specific needs, EMDESK's high cost and complexity make it less ideal for the Tartu City Government. A tailored, web-based solution focusing on budget and work time tracking with real-time updates and user-friendly interfaces would better serve their requirements. This solution would ensure effective management of multiple projects, precise tracking of expenses, and efficient use of financial resources, thereby supporting the city's goal of optimising project oversight and resource allocation.

3 Requirements

This paragraph introduces the Requirements Analysis, a crucial phase that defines the Functional Requirements, Non-Functional Requirements, and Prototype of the web application. Functional Requirements outline specific capabilities and behaviours necessary to fulfil user needs, while Non-Functional Requirements encompass criteria such as performance and security standards. Additionally, the section covers the development of a Prototype, providing a preliminary model to validate key features and interactions before full implementation. Together, these components ensure the application aligns with stakeholder expectations and effectively addresses user requirements.

3.1 Functional Requirements

Functional requirements (FR) are defined as the behaviour of the intended system, which can be represented through a precise and mathematical functional description [20]. During a client meeting focused on developing a web application, the following functional requirements were identified:

1. The system should allow *Users* to manage their accounts, including account creation, login, logout functionality and password restoration via email address.
2. The system should support two types of *User* roles: *Project Manager* and *Employee*.
3. The system should allow *Employees* to define, update, and delete their salary information.
4. The system should allow *Employees* to register completed work, specifying date, time spent, and linking it to a work package.
5. The system should allow *Employees* to register completed work based on a calendar.
6. The system should allow *Employees* to view their registered completed works.
7. The system should allow *Employees* to generate work hour reports for themselves covering various time periods.
8. The system should allow *Project Managers* to access a dashboard view where they can view both the projects they have initiated and those to which they have been assigned.

9. The system should allow *Project Managers* to create new projects and update existing ones, adjusting essential details such as project name, description, start/end dates, financial budget, and types of expenses as required.
10. The system should allow *Project Managers* to assign *Employees* to projects, allocate workload, and remove *Employees* from projects as needed.
11. The system should allow *Project Managers* to create, view and update work packages, including essential details such as name, description, person-month budget, start date, and end date.
12. The system should allow *Project Managers* to create, view, update, and delete expenses related to work packages, specifying essential details such as type, description, sum, and date of expense.
13. The system should allow *Project Managers* to access project information on person-months, salary, and expenses, including a summary that compares allocated budgets with current employee hour utilisation and monitors planned versus actual person-months and financial budgets.
14. The system should allow *Project Managers* to view a detailed overview of person-months, salary, and expense budgets organised by work package.
15. The system should allow *Project Managers* to register, view, modify, and delete completed work for employees and projects.
16. The system should allow *Project Managers* to view Employees, their respective workloads, work history, and the projects to which they are assigned.
17. The system should allow *Project Managers* to generate work hour reports for Employees covering various time period

3.2 Non-Functional Requirements

Non-functional requirements (NFR) specify the constraints and qualities the system must adhere to, often focusing on aspects beyond specific behaviours [20]. During the planning phase of a web application development project, attention was given to the following non-functional requirements:

Usability

1. The system should allow users to access the application over the internet via a desktop browser to ensure easy adoption and everyday use.
2. The system should display dates as *DD.MM.YYYY* and times as *HH:mm*, ensuring consistent formatting across all displayed dates and times.
3. The system should handle the simultaneous entry of work hours for multiple employees on a specific project, reducing the need for entering the data one by one.
4. The system should be easy to use without extensive training.
5. The user interface should be logical and intuitive, ensuring a smooth user experience.

Performance

6. The system should support multiple users working simultaneously, making information accessible to all users without performance degradation.
7. The system should be compatible with popular desktop browsers (e.g., Chrome, Firefox, Safari) to ensure a seamless and consistent user experience across different platforms.

Security

8. The system should ensure that project information is visible only to assigned individuals, protecting all project data.

3.3 Scenarios

This section describes the general scenarios in which the budget monitoring software will be used, illustrating typical interactions between the users and the system. These scenarios highlight the specific requirements and functionalities of the application to ensure it meets the needs of the Tartu City Government.

The primary objective of the budget monitoring software is to enable project managers to handle multiple projects simultaneously. Each project includes detailed budget, work time tracking, persons-months, labour costs, and expenses sections, allowing managers to oversee all aspects of their projects in one integrated platform.

A project manager logs into the system, creates new projects, enters initial budget, persons-months, labour cost estimates, and assigns employees to specific tasks. Employees then log their work hours and related expenses against these tasks daily, and the system provides

real-time updates on the budget, time usage, and expenses. This aggregated data helps the project manager monitor progress and update the project status as needed.

A critical function of the software is to track budget, work time, persons-months, labour costs, and expenses accurately for each project. This involves setting up various budget categories such as labour costs, travel expenses, and miscellaneous costs. Employees input their work hours and related expenses daily with date precision, and the system adjusts forecasts based on this actual data. For example, after an employee logs work hours and expenses, the system updates the project's budget, persons-months, labour costs, and time usage, providing real-time feedback that allows the project manager to ensure the project stays on track.

The software also enables the association of work hours with specific work packages. Project managers define work packages, and employees log their work hours against these predefined packages. This ensures accurate tracking of time spent on different tasks within a project. As employees log their hours, the system updates the status of each work package and calculates the corresponding persons-months, allowing project managers to review detailed reports and assess progress.

Reporting and forecasting are essential features of the software, enabling project managers to generate detailed reports and projections. The system allows managers to select projects and desired report types, generating reports that display current and forecasted data on budget, persons-months, labour costs, and expenses. Additionally, the system generates reports based on user activity, detailing the hours users spent on different projects and work packages, as well as total hours worked within specified date ranges. This functionality supports informed decision-making by providing insights into project performance, budget utilisation, and resource allocation.

User identification and access management are crucial for secure system access. Users must register and login to access the software, with the system distinguishing between employees and project managers. This role-based access control ensures that employees see only the relevant data, while project managers have the necessary permissions to oversee and manage all aspects of the projects. For example, a user registers themselves and assigns their role, after which they log in and access functionalities based on their assigned role.

These scenarios provide a comprehensive overview of how the budget monitoring software will be used in practice. They ensure that the software is tailored to the real-world requirements of its

users, facilitating effective project management and budget monitoring for the Tartu City Government.

4 Development

The development of the budget monitoring software involved carefully selecting modern tools and technologies to ensure a robust, scalable, and efficient application. This section details the primary technologies used, their features, advantages, and the rationale behind their selection.

4.1 Development tools and technologies

In developing the budget monitoring software, various modern tools and technologies were employed to ensure a robust, scalable, and efficient application. This section outlines the primary technologies used, highlighting features, advantages, and the rationale behind selection.

4.1.1 JavaScript - React

According to Banks and Porcello [26], React is a JavaScript library for building user interfaces, particularly single-page applications. Developed by Facebook, React emphasises the creation of reusable UI components. One of its standout features is the virtual DOM, which enhances performance by minimising direct manipulation of the browser's DOM.
Banks and Porcello also note [26], that React has a large community and a vast ecosystem of third-party libraries and tools, which can be advantageous for developers looking for extensive support and resources. However, React's learning curve can be steep due to its reliance on JSX and the need to understand the intricacies of state management.
Despite its strengths, React was not chosen for this project primarily due to the author's greater familiarity with Vue and the project's small team size, which did not necessitate the extensive resources available in the React ecosystem.

4.1.2 JavaScript - Angular

According to Seshadri and Green [27], Angular is a comprehensive JavaScript framework developed by Google for building dynamic web applications. Unlike Vue and React, Angular is a full-fledged framework that provides a complete solution, including a powerful template system, dependency injection, and end-to-end tooling. They note that

Angular's robust architecture and built-in functionalities make it suitable for large-scale applications. However, its complexity and steep learning curve can be challenging for smaller teams or projects requiring quick development cycles.
Angular was considered but ultimately not selected due to its complexity and the project's need for a more straightforward solution. The team's familiarity with Vue also significantly influenced this decision.

4.1.3 JavaScript - Vue

According to Vue.js official documentation [21], Vue is a JavaScript framework for building user interfaces and single-page applications. It is designed to be incrementally adaptable, focusing on the view layer only, and can be easily integrated into projects or libraries. Vue is highly regarded for its simplicity, flexibility, and performance, making it a popular choice for creating modern web applications.
As outlined in the Vue.js official documentation [21], one core feature of Vue is its reactive data binding system, which automatically syncs data between the model and the view, simplifying UI management and reducing boilerplate code. Vue offers a component-based architecture for creating reusable components with their own HTML, JavaScript, and CSS, promoting modularity and maintainability. According to Varun Bhagat article [22], the VueJS ecosystem includes Vuex for centralised state management and Vue Router for single-page application navigation, both of which enhance productivity and code quality.
However, Vue also has some drawbacks. Its relatively smaller community compared to frameworks like React or Angular can result in fewer third-party libraries and resources [23]. Additionally, Vue's flexibility can result in inconsistent code structures, particularly in larger teams [24].
Vue.js was selected by the author for the first version of the software due to their familiarity with Vue technology, its component-based approach, and the small team size. These factors make Vue an ideal choice for developing dynamic and responsive web applications.

4.1.4 Tailwind CSS

According to Tailwind CSS official documentation [28], this utility-first CSS framework that emphasises simplicity and flexibility, offering a wide range of utility classes for styling

web interfaces without predefined components. It enables developers to rapidly create responsive layouts by composing these classes, reducing the need for custom CSS and maintaining a scalable codebase.

However, mastering its utility classes initially requires some learning, and the resulting HTML markup can be more verbose than traditional CSS frameworks.

Despite these considerations, Tailwind CSS was chosen for its efficiency in quickly building custom, responsive designs tailored to the project's requirements, leveraging reusable components for consistent and efficient development.

4.1.5 Amazon Web Services

According to Sosinsky [29], AWS provides a comprehensive suite of cloud computing services, including computing power, storage options, and various managed databases. It offers high scalability, reliability, and various services tailored to different application needs. While AWS is a robust platform for building scalable and high-performance applications, its complexity and cost can be prohibitive for smaller projects or teams. Additionally, integrating other services can sometimes require significant configuration and management efforts.

AWS was considered but not chosen in favour of Google Cloud Platform (GCP) and Firebase, which offer more seamless integration for the specific needs of this project, including real-time data synchronisation and simpler backend management.

4.1.6 Google Cloud Platform

Google Cloud Platform (GCP) offers [46, 47] many cloud computing services for efficiently developing, deploying, and scaling applications. It includes computing, storage, databases, machine learning, and more, allowing developers to leverage advanced technologies without managing infrastructure. GCP's global scalability and integration with Google services like Firebase make it ideal for building flexible, high-performance applications.

4.1.7 Firebase SDK

According to the official Firebase documentation [30], the Firebase SDK (Software Development Kit) offers a powerful set of tools and libraries for integrating Firebase services into mobile and web applications effortlessly. It supports real-time databases, authentication, cloud storage, and more, enabling developers to build responsive apps with features like offline support and secure user authentication. While it requires familiarity with its APIs, Firebase SDK streamlines backend development and is ideal for applications needing real-time updates and robust user management.

4.1.8 Cloud Firestore - NoSQL Database

According to the official Google Cloud documentation [33], the term NoSQL, short for "not only SQL", refers to non-relational databases that use a non-tabular format to store data, rather than in rule-based, relational tables like relational databases. NoSQL databases use a flexible schema model that supports a wide variety of unstructured data, such as documents, key-value pairs, wide columns, graphs, and more. Organisations choose NoSQL databases for their flexibility, horizontal scalability, and ease of development.

The official Google Cloud documentation [33] describes Firebase NoSQL as a cloud-based database with Cloud Firestore being a versatile and scalable solution designed for mobile, web, and server development. It facilitates real-time data synchronisation across client applications, supports offline functionality, and integrates seamlessly with various Firebase and Google Cloud services, including Cloud Functions.

However, Cloud Firestore also has some drawbacks. Official pricing documentation indicates [32] that one of the main concerns is its pricing model, which can become expensive as the application scales, especially when read and write operations increase significantly. Richman, J. notes [30], that while Cloud Firestore offers flexibility, it can lead to complex querying and indexing, making it harder to manage and optimise performance for complex data retrieval operations. Mohammed Kareem also found [34] that another issue is vendor lock-in, as applications built using Cloud Firestore are tightly integrated with Google's ecosystem, potentially complicating future migrations to other platforms.

The author chose this technology because, given the client's usage of up to 50,000 document reads per day and storing data up to 1 GiB, the program qualifies for the free plan [32, 35]. This decision was further supported by Firebase's flexibility and scalability, making it ideal for developing the initial version of the software.

4.1.9 MySQL

Official MySQL documentation describes [36] MySQL as a relational database management system known for its reliability, ease of use, and performance. It is particularly suitable for applications requiring complex queries and transactions. MySQL offers strong support for data integrity and ACID (Atomicity, Consistency, Isolation, Durability) compliance, making it a robust choice for many web applications. According to DuBois [48], managing and maintaining the MySQL database server can add overhead, especially for smaller teams. MySQL's capabilities extend to supporting multiple threads, handling large databases with billions of rows, and ensuring secure, networked access. These features make it an ideal candidate for businesses and individuals seeking a high-performance, open-source Relational Database Management System solution.
Despite its strengths, MySQL was not chosen for this project due to the benefits offered by a NoSQL database like Cloud Firestore, particularly its real-time data synchronisation and flexibility in handling unstructured data.

4.1.10 PostgreSQL

According to the official PostgreSQL documentation [37], PostgreSQL is another relational database management system known for its advanced features, including support for JSON data types, full-text search, and robust transactional capabilities. It is highly extensible and offers excellent performance and scalability. PostgreSQL's ability to handle complex queries and data structures makes it a strong contender for many applications. However, similar to MySQL, the need for server management and maintenance can be a drawback for smaller teams.

Momjian elaborates [49] that PostgreSQL’s architecture supports complex operations and large-scale data management efficiently, making it a preferred choice for applications

requiring intricate data handling and high reliability. His discussion on subqueries highlights the database's capability to perform nested queries and other advanced SQL operations, enhancing its flexibility and power in handling diverse data requirements.

PostgreSQL was not selected due to the specific advantages of Cloud Firestore for this project, such as its seamless integration with Firebase services and real-time data handling capabilities.

4.1.11 Firebase Security Rules

Firebase Security Rules provide [38] a mechanism to control access to data stored in Firebase and ensure that it remains protected from unauthorised access. These rules are crucial in managing the permissions for reading and writing data within Firebase products such as Cloud Firestore. Firebase Security Rules operate as a flexible, expression-based syntax that can be customised to enforce specific security protocols and access control measures.

Firebase Security Rules are enforced [39] on the server side, meaning that they are executed regardless of the client application’s state. This server-side enforcement ensures that the data remains protected even if the client application is compromised. These rules enable developers to define who has access to which data and under what conditions, thereby supporting the implementation of fine-grained security policies.

4.1.12 Firebase Hosting

Firebase Hosting provides [40] developers with a robust platform for deploying web applications and static content effortlessly. It offers scalability, security, and reliability, ensuring fast access for users worldwide through a global content delivery network (CDN).

Firebase Hosting seamlessly integrates [40] with other Firebase services, simplifying full-stack application deployment. It supports custom domain hosting, enhancing credibility and visibility. With a powerful CLI tool, developers can streamline deployment, manage updates effectively, and utilise features like version rollback and preview deployments.

Moreover, Firebase Hosting offers a compelling advantage with its no-cost plan, making it accessible for developers to deploy and host their applications without incurring additional

expenses [35]. This affordability, coupled with its comprehensive features, makes Firebase Hosting an ideal choice for deploying the first version of the software efficiently and securely, ensuring a seamless experience for users.

4.1.13 Firebase Authentication

Firebase Authentication [41] is a robust authentication solution that enables developers to integrate secure user authentication into their applications easily. It offers a variety of authentication methods, including email/password authentication, phone number authentication, social authentication using providers like Google, Facebook, and Twitter, and custom authentication using external identity providers.

Firebase Authentication handles [41] all aspects of user management, including user registration, password recovery, and account verification. It provides a secure infrastructure for storing and managing user credentials, ensuring that sensitive information is encrypted and protected from unauthorised access.

Firebase Authentication also offers [42] built-in support for user session management, allowing developers to easily implement features such as persistent login sessions and automatic session expiration. This simplifies the development process and improves the overall user experience by reducing the need for manual session management code.

The author opted for email/password authentication as the primary method for user authentication, ensuring secure access while simplifying the process. Firebase Authentication provides a seamless, scalable solution, integrating smoothly with other Firebase services for comprehensive user management.

4.2 Application Architecture

The architecture of the web application is illustrated in Figure 6, which centres around a front-end architecture built using Vue.js with Tailwind CSS. This setup enables a responsive and dynamic user interface experience. The application communicates directly with a Firebase NoSQL database, specifically Cloud Firestore, which serves as the data storage and management system in JSON format. Firebase Security Rules are implemented to protect the database, ensuring that access is granted only to authorised users based on

predefined criteria. This architecture emphasises a serverless approach, leveraging Firebase's capabilities for real-time data synchronisation and seamless scalability without relying on a traditional backend infrastructure.

The architecture of a Vue.js web application operating without a traditional backend leverages Google Cloud Platform services to provide essential backend functionalities. Cloud Firestore serves as the NoSQL database, handling data storage and retrieval with support for real-time synchronisation and efficient querying through Firebase SDK integration with Vue.js components. Firebase Security Rules enforce access control policies based on user authentication and specific conditions, ensuring secure data access. Firebase Hosting hosts the frontend components of the Vue.js application and static assets, providing fast global delivery. Firebase Authentication manages user authentication and authorization, supporting various login methods and securely handling user sessions. This architecture enables developers to focus on frontend development while leveraging scalable infrastructure and robust backend services provided by Google Cloud Platform, facilitating secure and efficient operation of Vue.js web applications.

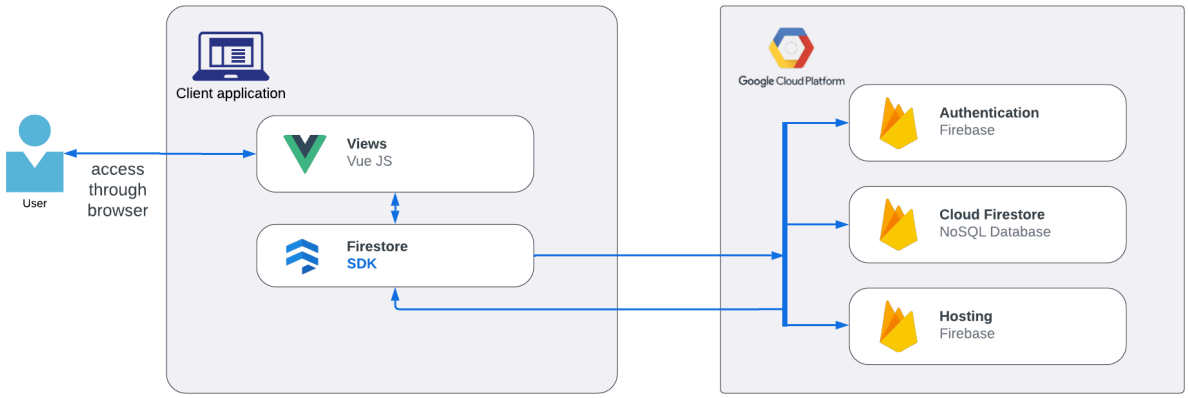


Figure 6. Application architecture

4.3 Database Architecture

This paragraph introduces the database architecture used in the system, focusing on the structure and organisation of data across different entities. Firestore, a NoSQL database, is utilised, employing collections and sub-collections to manage and store data effectively. Collections function similarly to tables in relational databases, while sub-collections act as

nested tables, allowing for hierarchical data organisation. This architecture supports the dynamic nature of the data and enhances query efficiency and data retrieval. The design choices and their benefits in meeting the application's requirements will be examined in detail.

4.3.1 User and Project relation

The database structure efficiently manages user and project-related data through the User, Project, Work History, and Expenses collections.

- *User*: Represents individual users. Each user can be assigned to or own multiple projects. Additionally, users can have multiple work histories and associated expenses, reflecting the work they complete and the costs they incur.
- *Project*: Represents individual projects. Each project can have multiple users assigned to it, indicating collaboration and ownership. This allows for detailed tracking of project assignments and user involvement.
- *Work History*: Records the activities related to specific projects and work packages. Users can have many work histories, which they register upon completing work. This ensures detailed tracking of user contributions and project progress.
- *Expenses*: Tracks the expenses incurred within a project or work package. Users can register multiple expenses, enabling comprehensive financial tracking and management of their work.

This structure ensures efficient management of user assignments to projects, work histories, and expenses, facilitating detailed oversight and collaboration tracking within the system. Figure 7 provides a visual representation of the database structure, including the collections for *Users*, *Projects*, *Work History*, and *Expenses*, illustrating the relationships between *Users* and *Projects*.

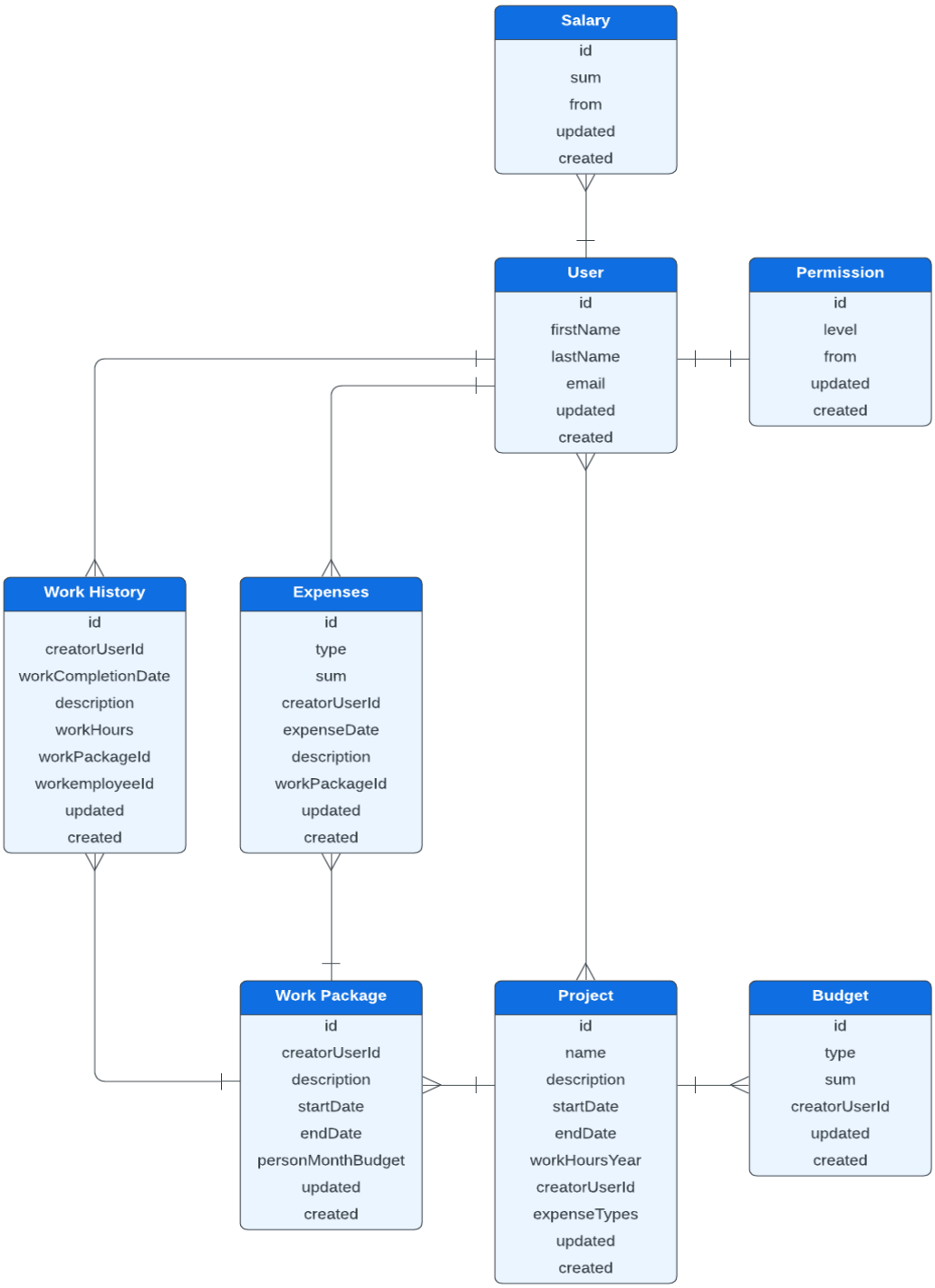


Figure 7. Relations between different entities.

4.3.2 User, Salary and Permission Structure

The database structure efficiently manages user-related data through *User, Salary,* and *Permission* collections.

- *User*: Represents individual users. The *User* collection stores detailed information about each user, providing a central repository for user data.
- *Salary*: A sub-collection under *User*. Each user can have multiple salary records because salaries can change over time. This sub-collection stores individual salary entries, detailing the amount and the period from which each salary applies, ensuring comprehensive historical salary data management.
- *Permission*: Manages the permission levels associated with each user. Each user has one associated permission, ensuring proper access control and role management within the system.

This structure ensures efficient user information management, their corresponding salary histories, and permission levels. Figure 8 provides a visual representation of the database structure, including the collections *Salary, User,* and *Permission.*

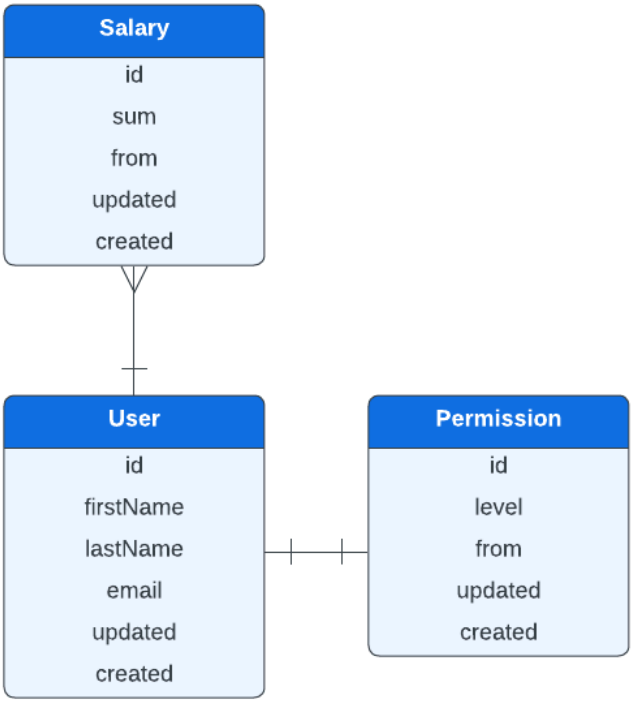


Figure 8. User, Salary and Permission relations.

4.3.3 Project, Budget, Work Package, Expenses and Work History Structure

The database structure efficiently manages project-related data through collections for *Project, Budget, Work Package, Expenses,* and *Work History*.

- *Project*: Represents individual projects. Each project can have multiple work packages and different budgets. This allows for detailed planning and tracking of various aspects of a project.
- *Work Package*: Represents distinct segments of work within a project. Each project can contain multiple work packages, facilitating the breakdown of the project into smaller, manageable parts. Each work package can have numerous expenses and work histories associated with it.
- *Expenses*: Linked to work packages, this collection tracks all expenses incurred within a work package. This structure allows for detailed financial tracking and management of project costs.

- *Work History*: Records the work activities related to specific work packages. Multiple work histories can be associated with a single work package, enabling detailed tracking of labour and progress within each project segment.
- *Budget*: Represents the financial allocations for projects. A project can have multiple budgets to manage different expenses and financial plans, ensuring comprehensive financial oversight.

This structure ensures efficient and detailed management of projects, their financial aspects, and the work performed within them. Figure 9 provides a visual representation of the database structure, including the collections *Project, Budget, Work Package, Expenses* and *Work History.*

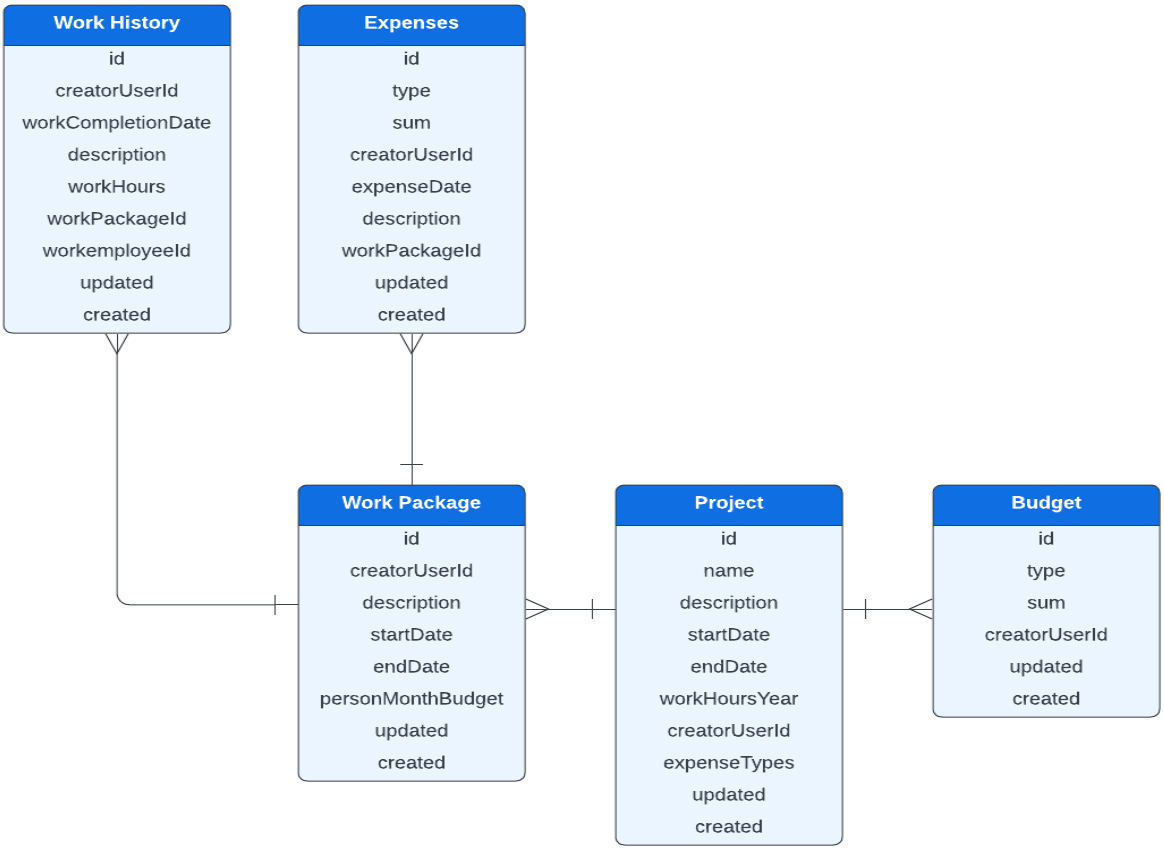


Figure 9. *Project, Budget, Work Package, Expenses* and *Work History* relations.

4.4 Business logic

The business logic of a project management system encompasses a range of calculations and processes essential for planning, tracking, and managing resources and costs effectively. Key components include Persons-Months (PM), labour costs, and expenses, each vital in ensuring the project remains on schedule and within budget. Project managers can make informed decisions and maintain control over the project's financial and operational aspects by accurately quantifying and aggregating the efforts and expenditures associated with various project tasks and resources. The following subsections provide detailed methodologies and formulas for calculating Persons-Months, used Persons-Months, labour costs, and expenses, offering a comprehensive framework for effective project management.

4.4.1 Persons-Months

In project management, Persons-Months (PM) is a crucial metric used to estimate the total effort required to complete a project. It quantifies the work done by team members over a period, expressed in person-months, and is essential for effective planning, budgeting, and progress tracking. PM represents the amount of work one person can complete in one month. It helps project managers allocate resources, forecast timelines, and manage workloads by considering the number of people involved and their working duration.

Persons-Months (PM) Calculation

The calculation of PM involves aggregating the work efforts of all individuals in a project over a specific period. It can be applied to tasks, work packages, or entire projects, providing a comprehensive view of labour requirements. The formula for PM is:

$$PM_{project} = \sum_{i=1}^{N} \left(U_i \times M_i \right)$$

Where:

N = Total number of tasks or work packages.

U_i = Number of users or personnel assigned to task

M_i = Number of months (or fraction of a month) task *i* takes to complete.

Using this formula, the total PM for a project is derived by summing up the individual contributions of the work package, considering the personnel involved and the duration of their involvement.

Used Persons-Months (PM) Calculation

Used Persons-Months (PM) calculation leverages the hierarchical data organisation provided by collections and sub-collections. Specifically, the Work History collection plays a crucial role in tracking users' time and effort on various project tasks. Each work history entry records the user, the work package or project they contributed to, and the time duration of their work. To compute the used Persons-Months (PM) for each project or work package, it is necessary to aggregate the work durations and personnel data stored in the User and Work History collections.

Given the detailed data in the collections, the calculation of used PM can be expressed as:

$$Used\ PM_{project} = \sum_{i=1}^{N} \left(\frac{WH_i}{WHM} \right)$$

Where:

N = Total number of work history entries associated with the project.

WH_i = Work hours recorded in the i^{th} work history entry.

WHM = Work hours in a month (conversion factor, typically the total work hours in a year divided by 12).

This formula aggregates the total work hours across all work history entries for a project and converts these hours into person-months using the work hours per month conversion factor. Note that each project may have different yearly work hours, which affects the conversion factor.

4.4.2 Labour Cost

In project management, labour cost calculation is a critical process that ensures that the financial resources allocated to personnel are effectively managed to meet project objectives. This involves estimating the total labour costs required to complete a project by calculating the cost of work done by each team member. Accurate labour cost calculation is essential for effective financial planning, resource allocation, and cost control, enabling project managers to monitor expenditures and stay within financial constraints.

Calculation

The labour cost calculation aggregates the costs associated with all work histories for the project. Each work history entry records the work done by an employee, and the corresponding cost is computed by multiplying the hours worked by the employee's hourly salary. The total labor cost is then determined by summing the costs of all work history entries. The formula for calculating the total labour cost is:

$$Total\ Labor\ Cost_{project} = \sum_{i=1}^{N} \left(H_i \times S_i \right)$$

Where:

N = Total number of work history entries for the project.

H_i = Number of hours recorded in the i^{th} work history entry.

S_i = Hourly salary of the employee associated with the i^{th} work history entry.

This formula aggregates the total labour costs across all work history entries to determine the overall labour cost needed for the project. Note that each project may have a different amount of yearly work hours, which affects the calculation of the hourly salary.

4.4.3 Expenses

In project management, calculating expenses is a crucial process to ensure that financial resources are allocated and utilised effectively to meet project objectives. This involves estimating and tracking all non-labor costs required to complete a project, including travel, materials, indirect, equipment, and miscellaneous expenses. Accurate expense calculation is essential for effective financial planning, resource allocation, and cost control, enabling project managers to monitor expenditures and stay within budgetary constraints.

Calculation

The expense calculation aggregates all costs associated with a project over its entire duration. Each expense entry records the cost incurred for a specific item or service, and these costs are summed to determine the total expenses for the project. The formula for calculating the total expenses is:

$$Total\ Expenses_{project} = \sum_{i=1}^{N} E_i$$

Where:

N = Total number of expense entries for the project.

E_i = Expense amount recorded in the i^{th} expense entry.

This formula aggregates the total expenses across all entries to determine the overall costs needed for the project.

4.5 General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) [43] is a comprehensive data protection law that governs the processing of personal data of individuals within the European Union (EU). Enforced since May 25, 2018, GDPR aims to protect the privacy and personal data of

EU residents, ensuring that their data is handled securely and transparently by organisations operating within or outside the EU.

4.5.1 Firestore Database Location

For the purpose of this thesis, the Firestore database is located in the European multi-region identified as "eur3" [44]. This geographical designation ensures that the data stored within this Firestore instance is physically located within the EU, thereby subjecting it to the stringent requirements of the GDPR.

4.5.2 Multi-Region Location Benefits

Selecting a multi-region location like "eur3" maximises database availability and durability. A multi-region location stores multiple replicas of the database across defined regions. These replicas can be either read-write, containing all the data, or witness replicas, participating in replication without holding all data.
This replication ensures data remains accessible even if an entire region is lost. Within a region, data is further replicated across zones to maintain availability despite zone failures.
For "eur3" [44]:

- *Read-Write Regions*: europe-west1 (Belgium), europe-west4 (Netherlands)
- *Witness Region*: europe-north1 (Finland)

This setup ensures data accessibility and durability, meeting high standards for availability and resilience.

4.6 User Roles and Permissions

The planned system will support two types of user roles: Employee and Project Manager. Each role will have specific permissions and access levels to ensure proper data security and enhance the user experience.

4.6.1 Employee

- Access to a personal dashboard that displays only the projects related to them.
- View their personal data, including salary information and registered work history.

- Option to add and manage completed work using a calendar interface.

4.6.2 Project Manager

- Full access to all projects within the system.
- Ability to create, update, and manage projects.
- Manage user data, assign employees to projects, allocate workloads, and view detailed project information.
- See reports on project progress and employee work hours, and manage project expenses.

This role-based access will ensure that Employees see only the data relevant to them, while Project Managers will have the necessary permissions to oversee and manage all aspects of the projects

5 Final Outcome

A new Web Application for Managing EU Projects was developed as part of this master's thesis. The completed application can be accessed at the link provided in Appendix 2, and the source code is available in Appendix 1.

The development of the budget monitoring software has resulted in a comprehensive and user-friendly system tailored to meet the needs of both Employees and Project Managers. The system incorporates various features and functionalities that ensure efficient project management, accurate tracking of work hours and expenses, and secure access to user data. This section outlines the key components and capabilities of the software, providing a detailed overview of user roles, session management, dashboards, and project-specific tools.

5.1 Session Management

The software includes comprehensive session management features to ensure secure and efficient user access.

5.1.1 Registration

New users can create an account by providing necessary information such as name, email, and password. The email address provided must be unique, preventing duplicate registrations. By default, the registration process sets the user type as Employee. New registered users will be granted Employee permissions. Upon successful registration, the user is redirected to the Employee dashboard view. Figure 10 provides a visual representation of the registration form used for user sign-up.

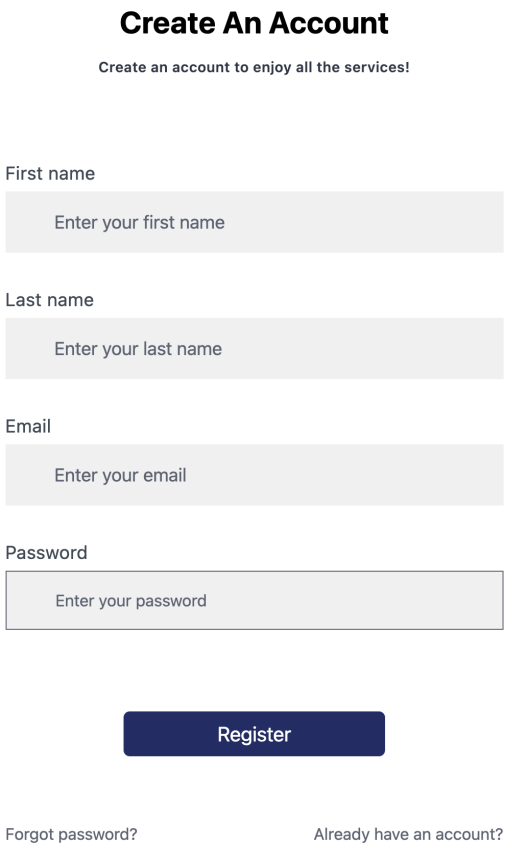


Figure 10. Sign-up form.

5.1.2 Login

Users can securely log in to their accounts using their registered email and password. Based on the user role, Employees are redirected to the Employee dashboard view, while Project Managers are redirected to the Projects dashboard view. Figure 11 provides a visual representation of the login form used for user sign-in.

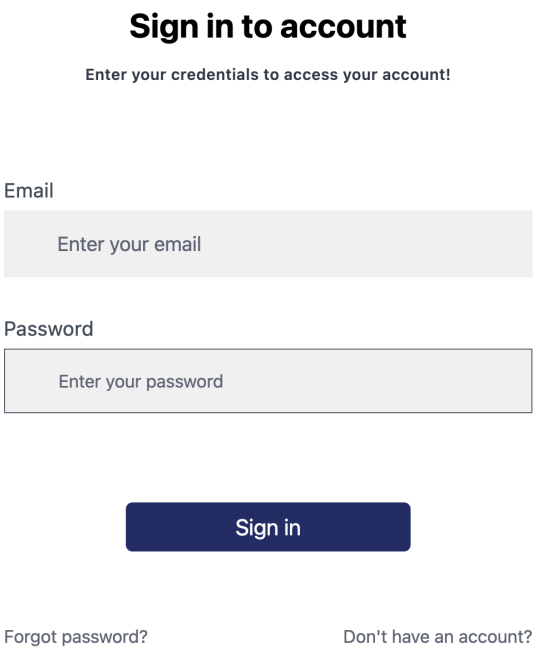


Figure 11. Sign-in form.

5.1.3 Password Restoration

Users who forget their passwords can initiate a password reset process. Below, Figure 12 provides a visual representation of the reset password form used for account restoration.

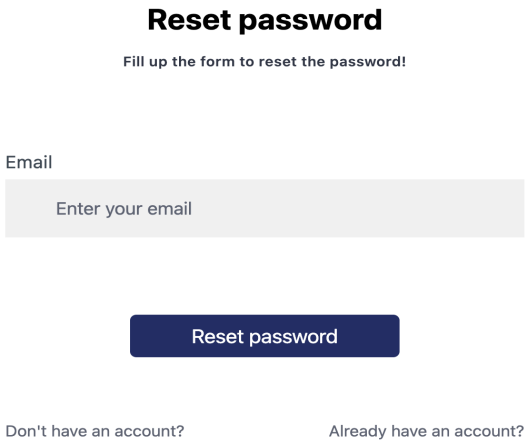


Figure 12. Password restoration form.

The system sends a password reset link to the user's registered email, allowing them to securely create a new password. Below, Figure 13 illustrates an example of the email template that users receive when restoring their password, which allows them to reset it.

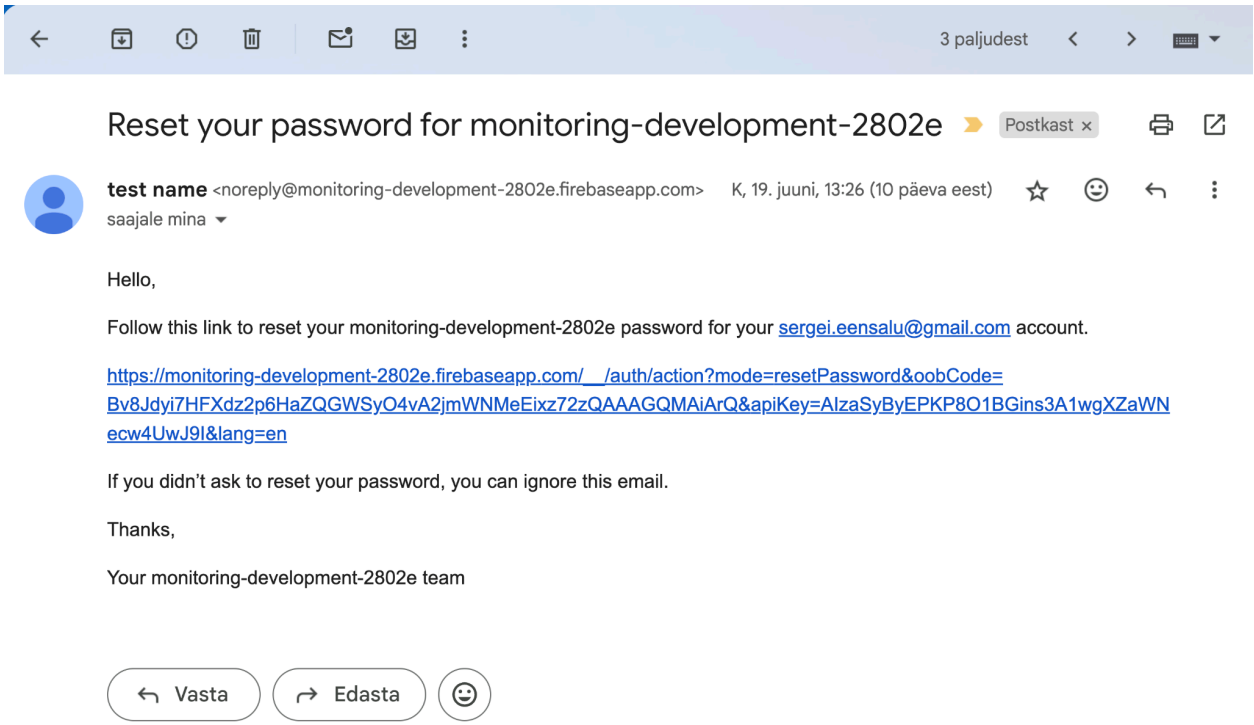


Figure 13. Password restoration email example.

5.2 User Dashboard

The user dashboard provides a centralised interface tailored to the needs of each user role. Employees and Project Managers have distinct dashboards designed to streamline their workflows and provide quick access to relevant information.

5.2.1 Projects and Workload

This view provides a comprehensive overview of all projects associated with the user. It also details the workload within each project, offering insights into task distribution and project demands. Figure 14 provides an example of a view that shows projects related to the user and displays the workload within each project.

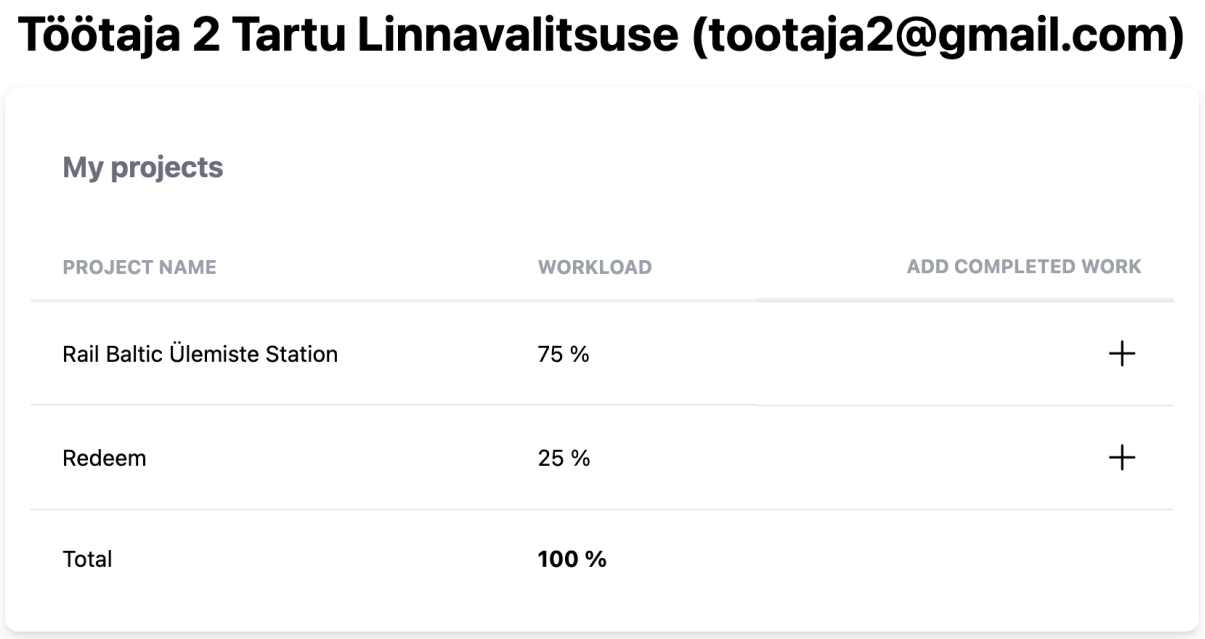


Figure 14. User projects.

This view enables users to add work history to specific projects via a calendar interface. Users can select a particular day on the calendar to view all registered work histories for that day, as illustrated in Figure 18. Additionally, users can click on any task to see its detailed view, which is shown in Figure 16. The interface also allows users to register new work history for a specific date by providing details such as a description, hours worked, and the associated work package, as depicted in Figure 17.

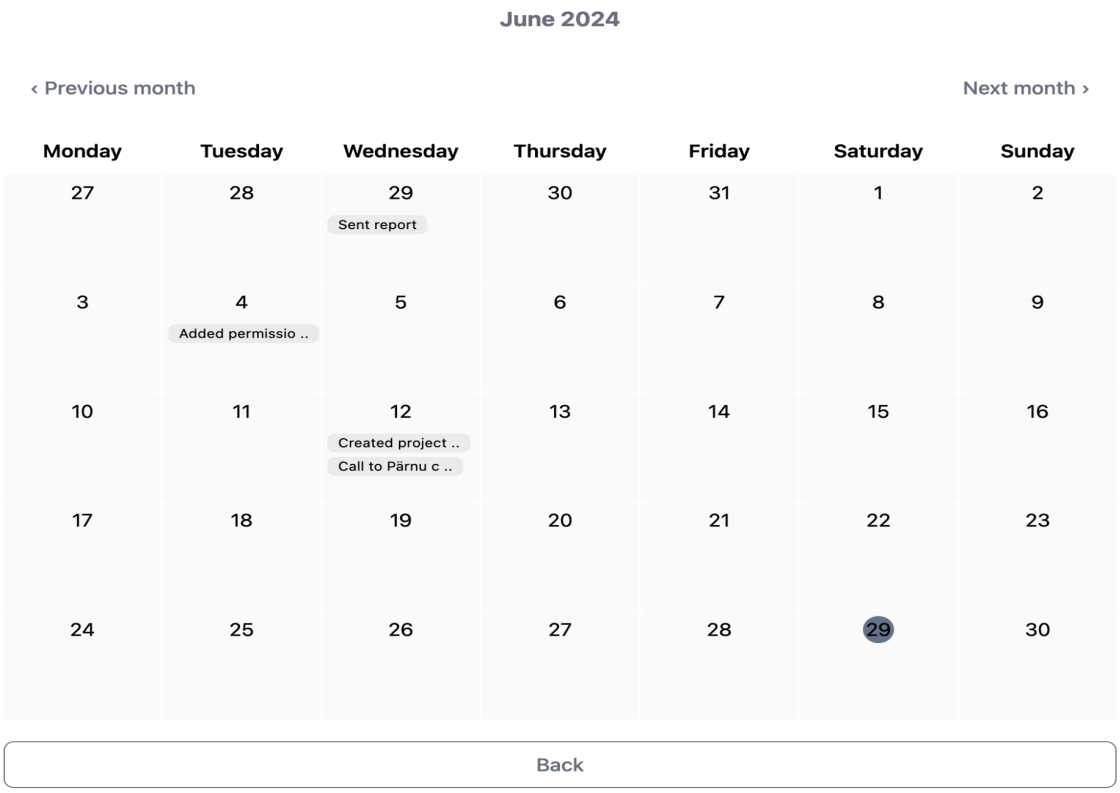


Figure 15. Calendar.

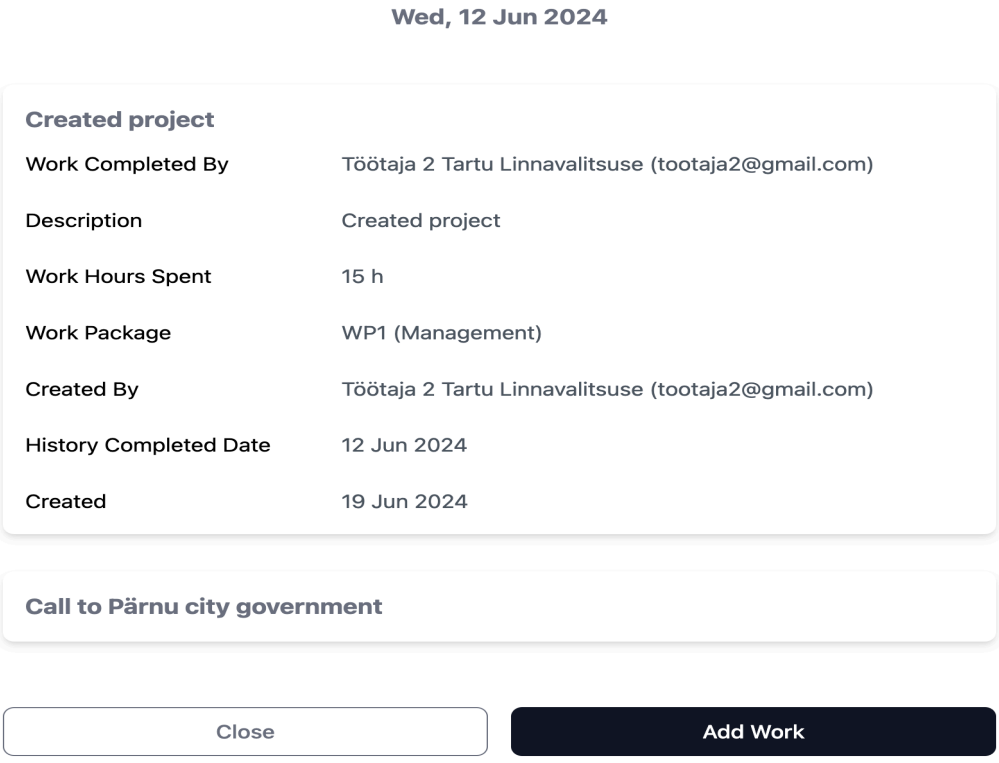


Figure 16. Work history details.

Figure 17. Work History creation.

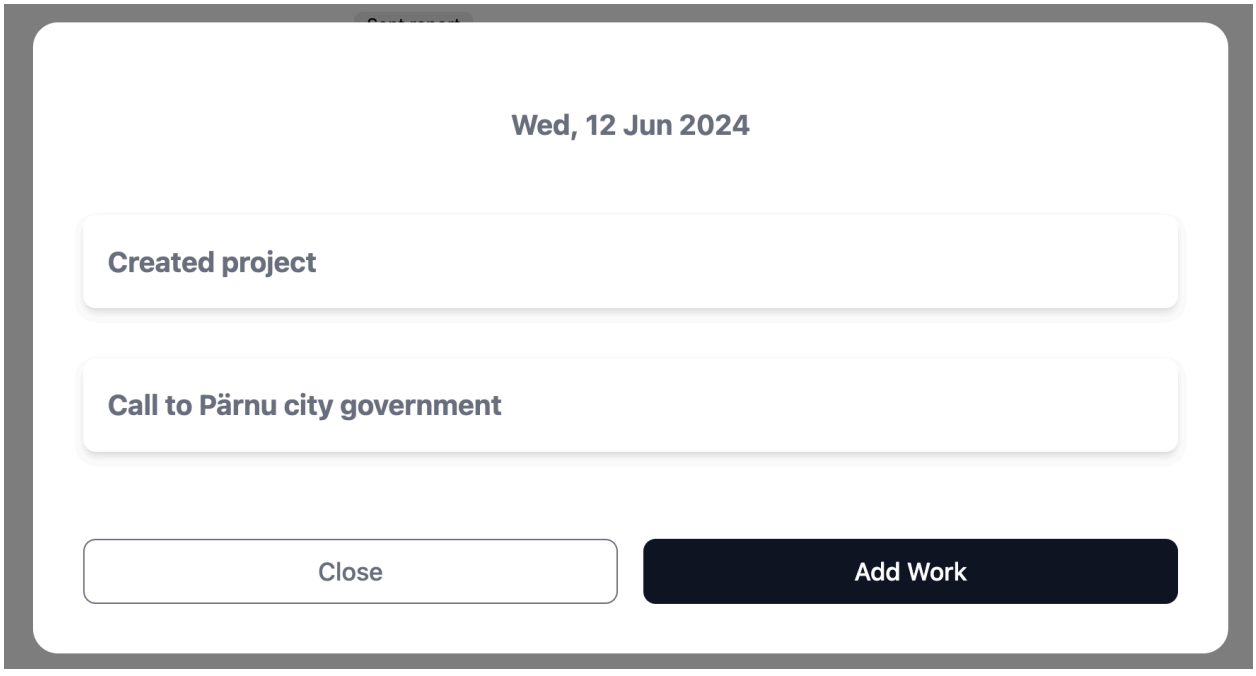


Figure 18. List of work histories on the selected date.

5.2.2 Salary

Employees can view their salary information, including historical updates, as illustrated in Figure 19. For example, it shows an employee's initial salary starting from January 1, 2024, with a subsequent increase effective from June 1, 2024. Project Managers have access to salary data for all employees involved in their projects. They have the ability to add new salary entries, edit existing ones, or delete salary information. Figure 20 demonstrates how salary information can be updated, while Figure 21 shows the confirmation modal for deleting salary information. Additionally, Figure 22 illustrates how to add a new salary entry, including specifying the salary amount and start date.

Figure 19. List of salaries.

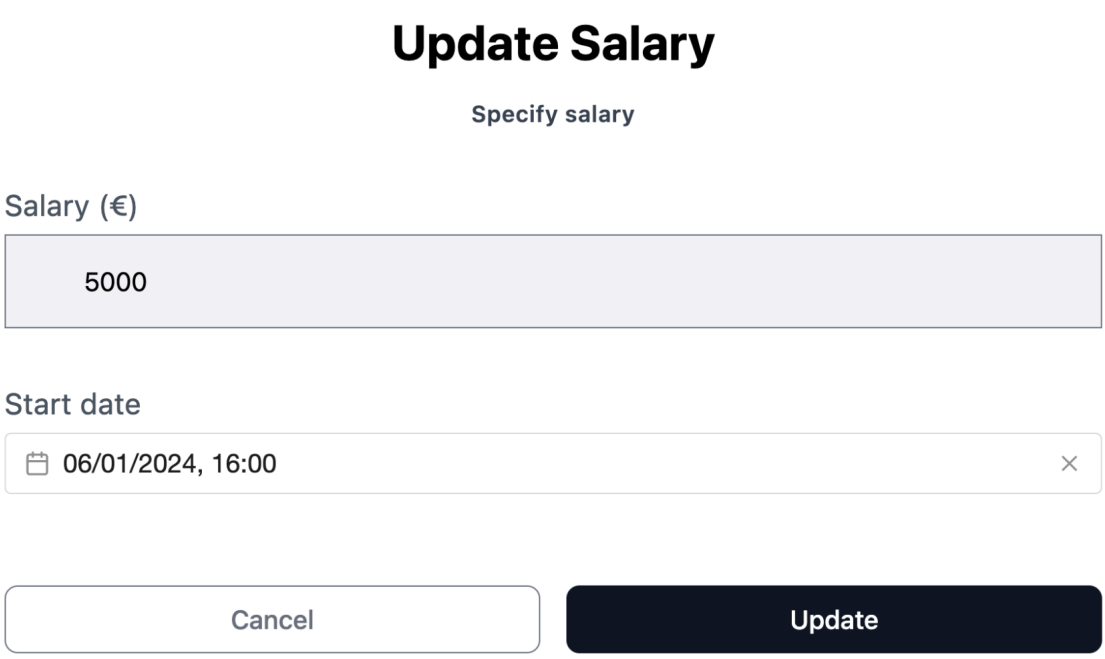


Figure 20. Edit salary view.

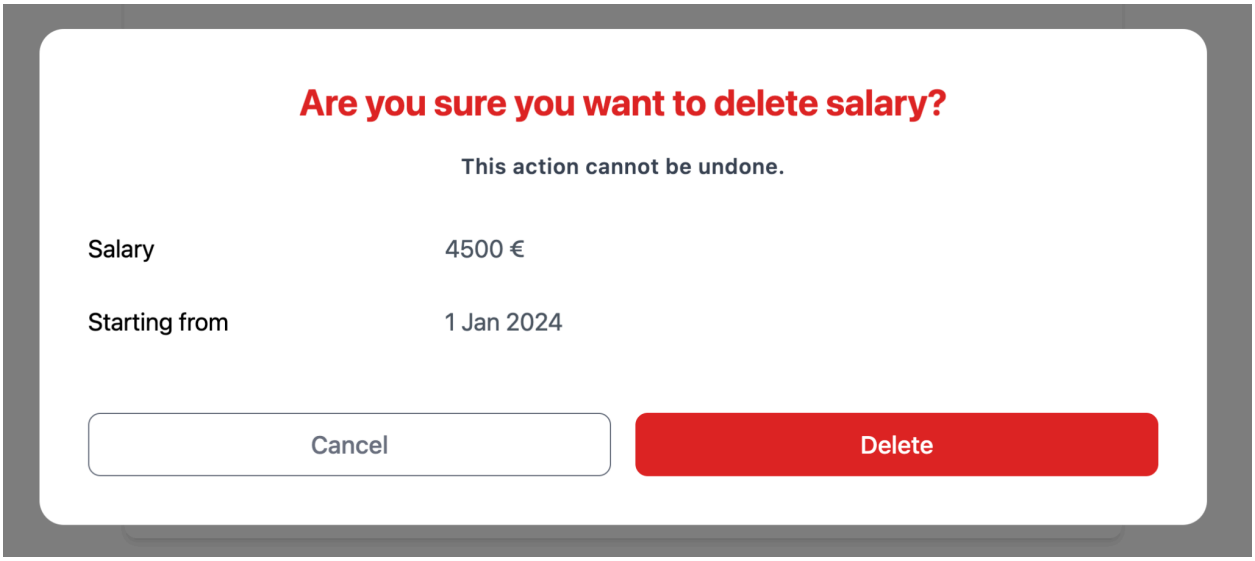


Figure 21. Delete salary view.

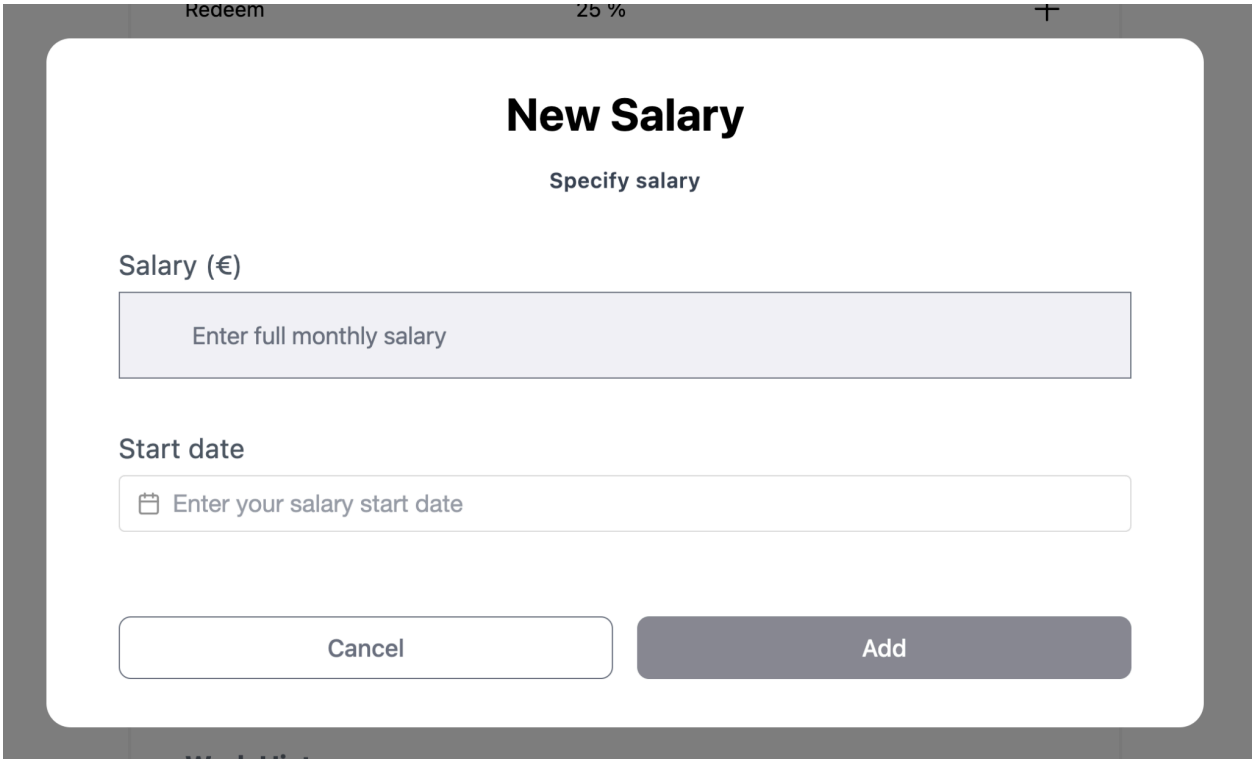


Figure 22. Create a new salary view.

5.2.3 Work History

Employees and Project Managers can view, update, and manage their work history, including completed tasks and logged hours. Figure 23 provides an example of the work

histories view, displaying details such as project names, work history descriptions, hours spent, and the dates when the work was performed.

Work History entries are editable, with fields such as work description, related work package, date of work, and hours spent being modifiable. Figure 24 illustrates an example of the work package edit view.

When deleting a work history entry, a confirmation modal is presented to ensure the user is aware of the action they are about to take. Figure 25 shows an example of this confirmation modal, which includes details of the work history to be deleted, providing a final overview before the action is finalised.

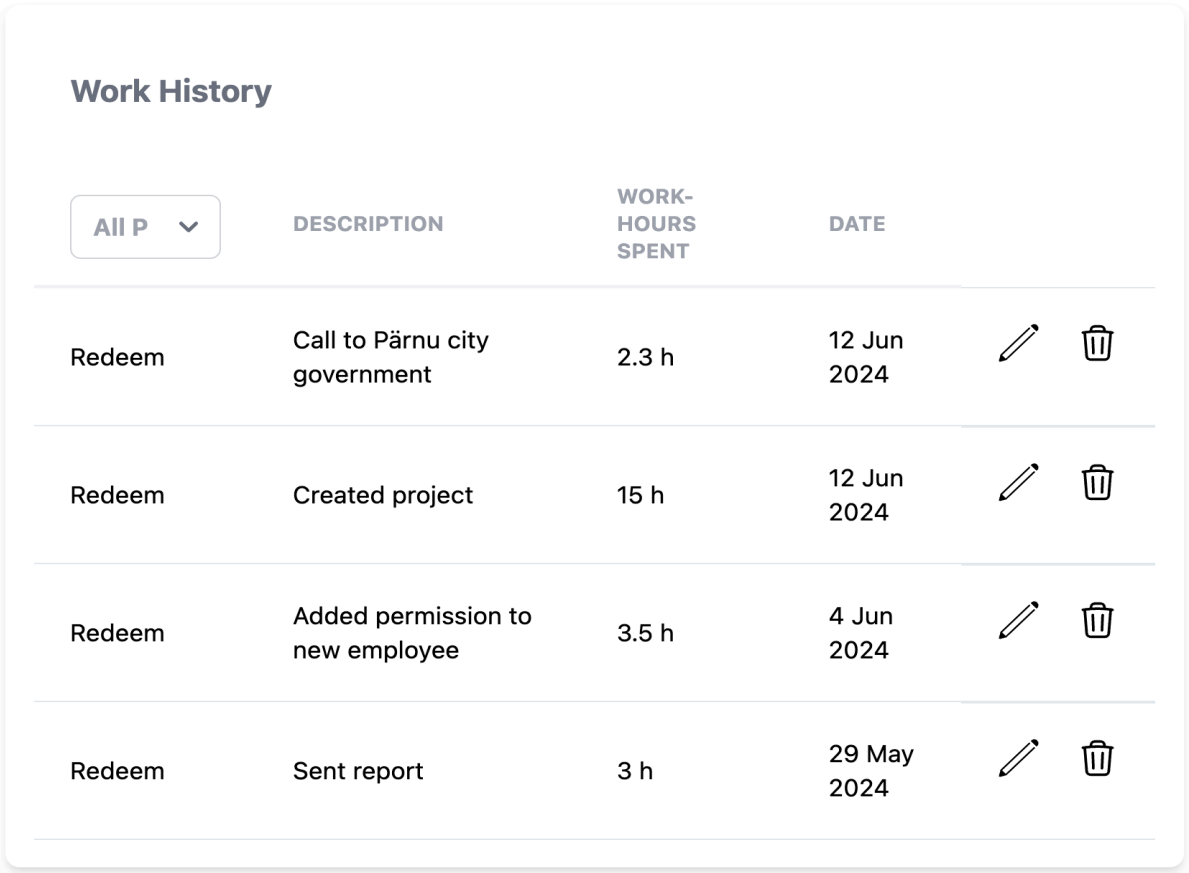


Figure 23. Work histories view.

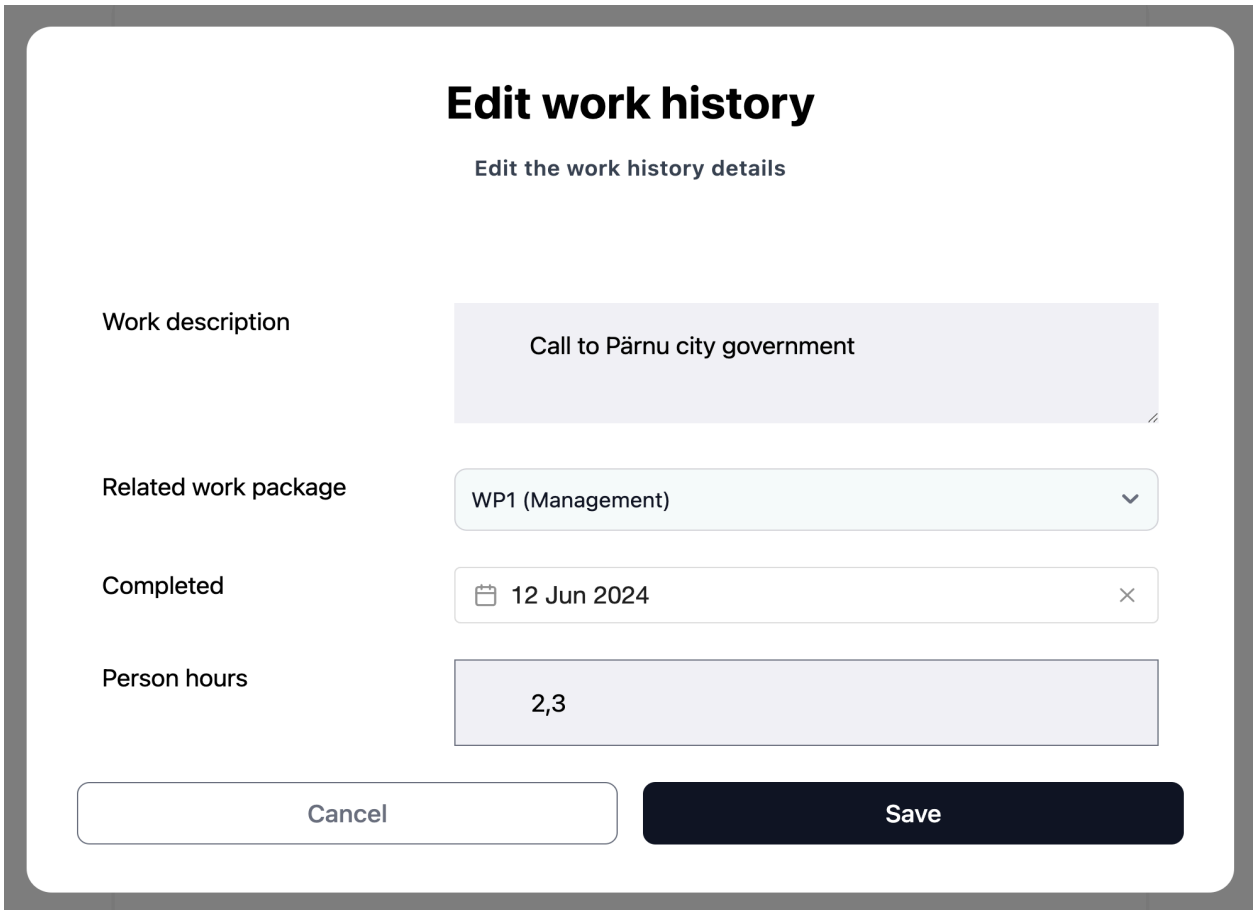


Figure 24. Work history update view.

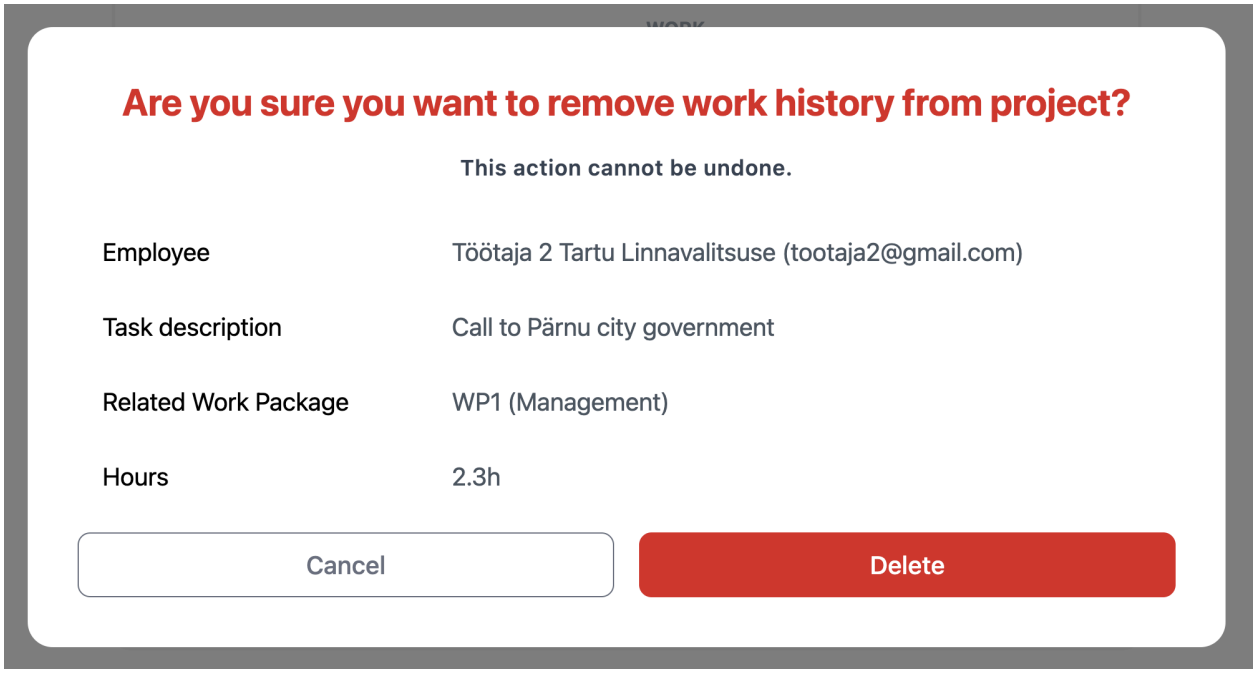


Figure 25. Work history delete view.

5.2.4 Reports

Employees and Project Managers have the option to generate detailed reports on work hours. They can specify the date range for the report, choosing from the current month, previous month, current quarter, current year, or a custom range. Reports can be generated in full detail or grouped by project or work package.

Figure 26 demonstrates one possible report configuration, where the filter is set to the current year. This example highlights all registered work histories, displaying work hours spent and grouping the data by project and work package. The report is sorted by the date when the work was done, with a summary showing the total hours allocated to each work package and project.

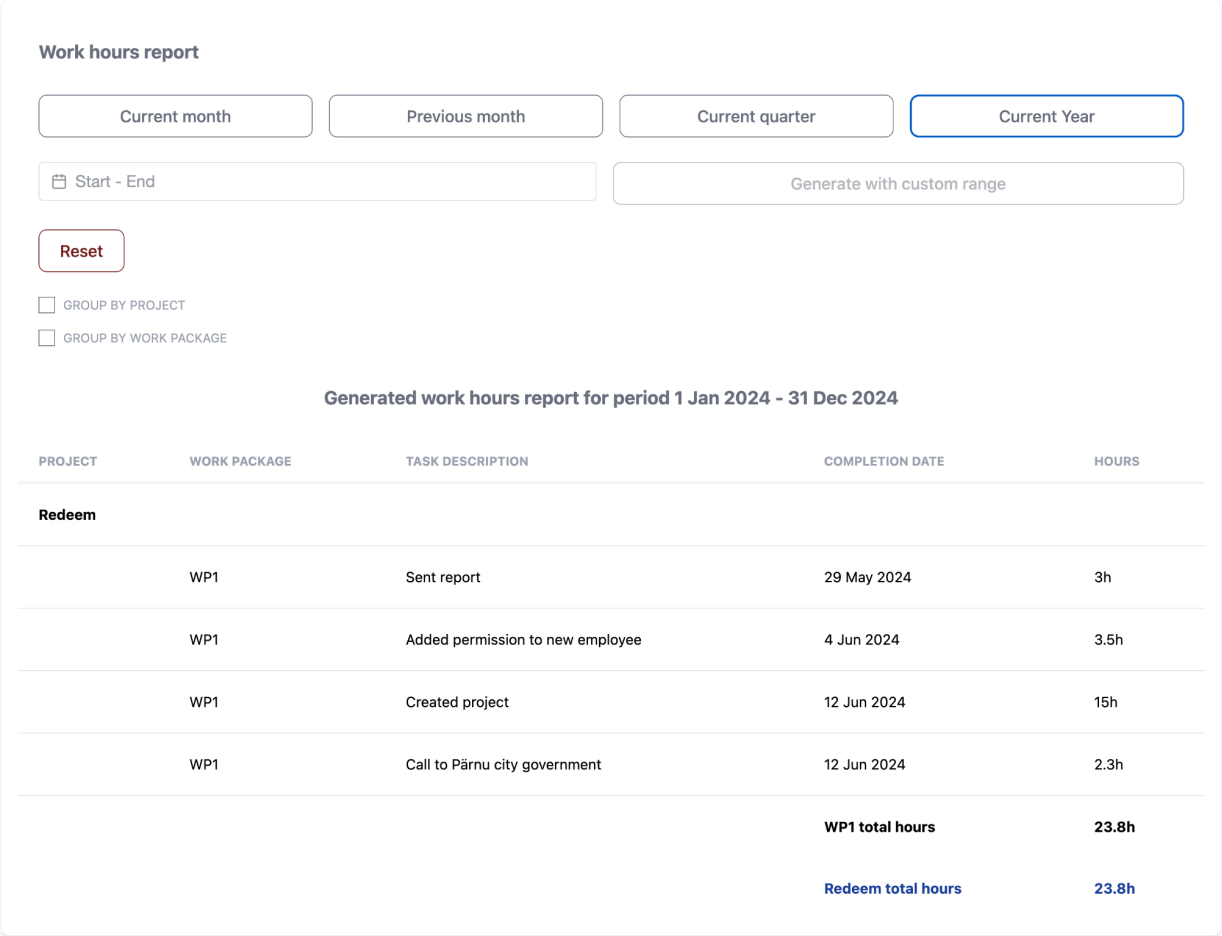


Figure 26. Work hours report for 01.01.2024 - 31.12.2024.

5.3 Project Dashboard

5.3.1 Details

Project Manager can access and update key project information, including project names, descriptions, start and end dates, and overall project status. Figure 27 provides an example of project details, showcasing project names, descriptions, start and end dates, overall project status, annual work hours, and expense types. Figure 28 demonstrates the project editing interface, where users can modify the project's budget, expense types, status, name, description, and start and end dates.

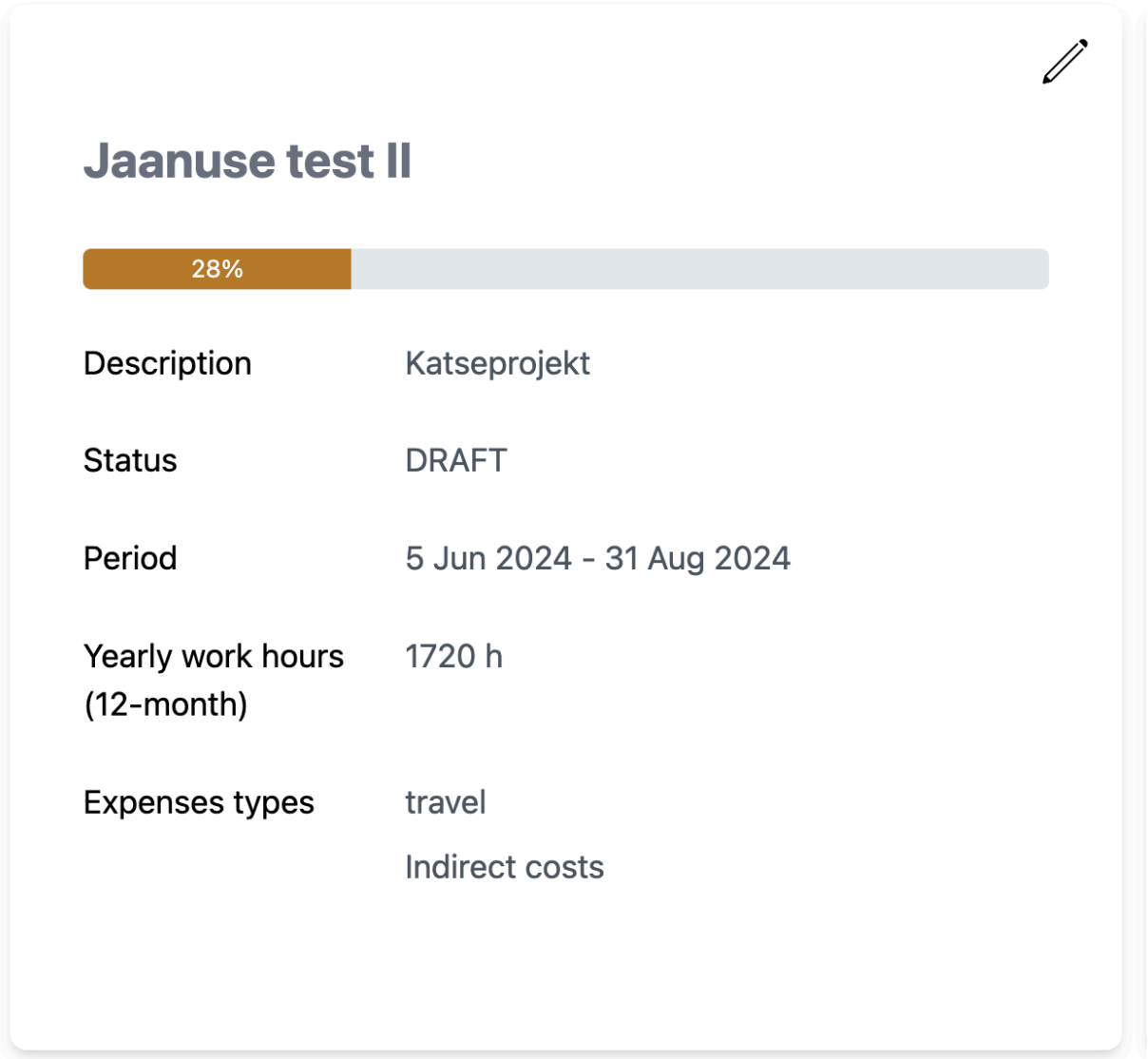


Figure 27. Project details view.

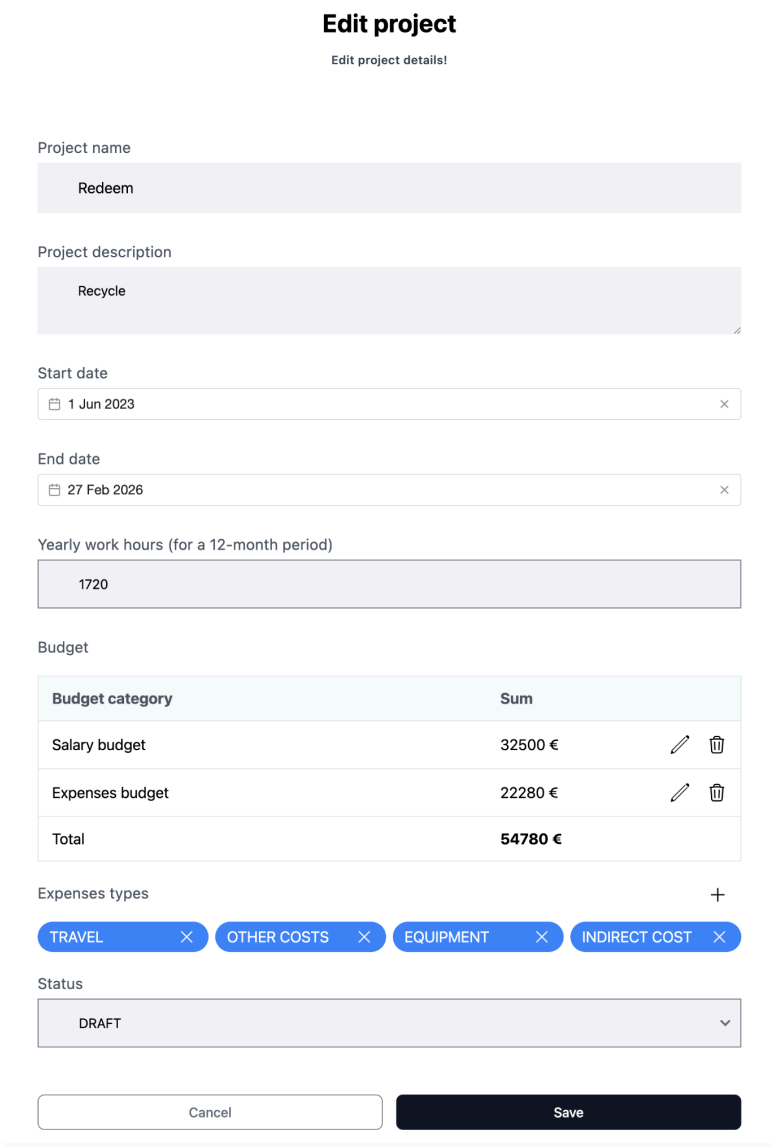


Figure 28. Project update view.

5.3.2 Budget Summary

The Budget Summary provides a comprehensive overview of the project budget, including person-months, salary, expenses, and hours allocated and utilised. Figure 29 illustrates how each budget category - person-months, salary, expenses, and hours - is displayed in separate columns, with *Used* and *Budget* values clearly distinguished for each category.

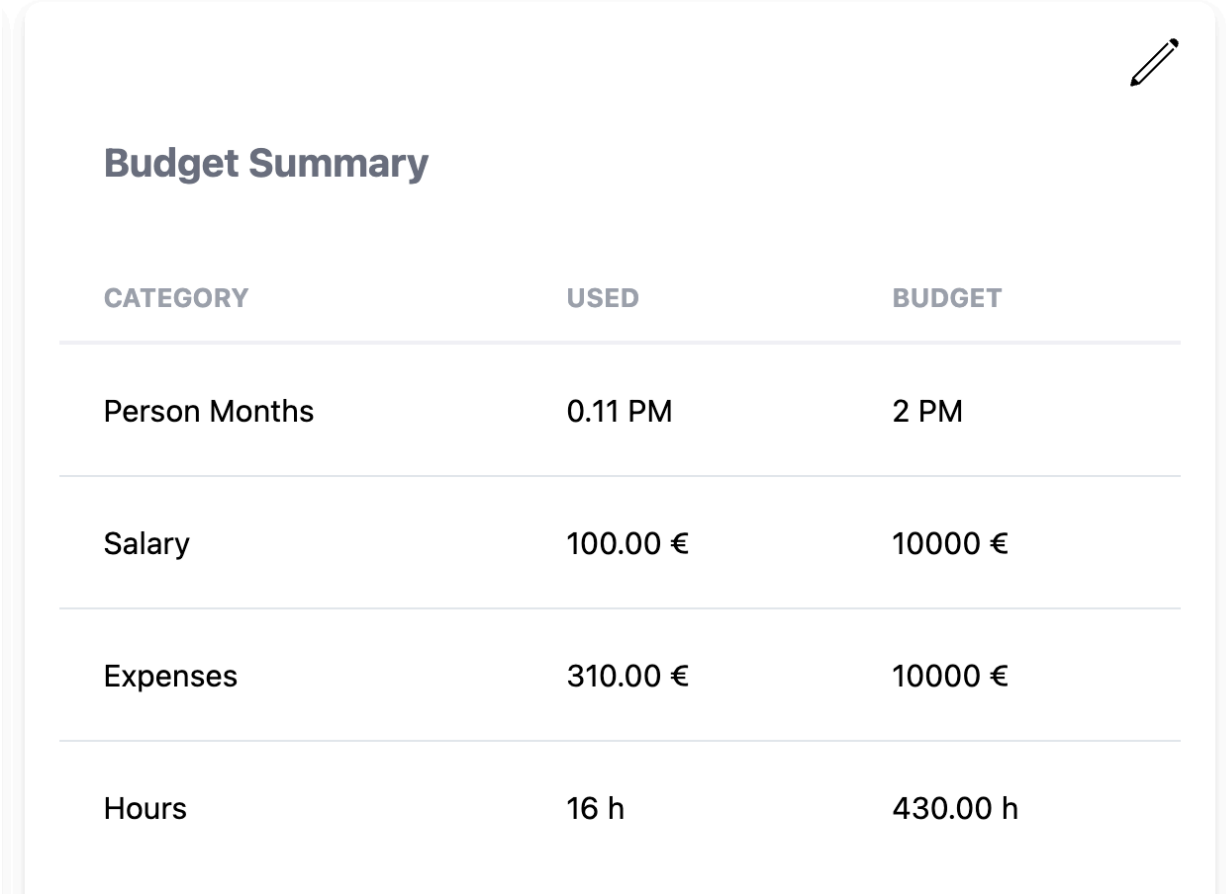


Figure 29. Budget summary view.

5.3.3 Overview of Budget Resources

The overview of budget resources presents a detailed breakdown of each category - person-month resources, salary budget resources, and expenses budget resources -organised by work package and further divided by month. Each section shows the utilised resources within each work package and month. Figure 30 provides a detailed view of person-month allocation and utilisation, while Figure 31 offers insights into salary budgets and expenditures. Figure 32 illustrates the breakdown of expenses by work packages.

Person Month Resources

	5 JUN 2024 - 30 JUN 2024		1 JUL 2024 - 31 JUL 2024		1 AUG 2024 - 31 AUG 2024		
WORK PACKAGE	USED / LEFT		USED / LEFT		USED / LEFT		USED / TOTAL PM BUDGET
WP1 (Admin) 5 Jun 2024 - 31 Aug 2024	0.06 PM / 0.28 PM		0.00 PM / 0.33 PM		0.00 PM / 0.33 PM		0.06 PM / 1 PM
WP2 (Communication) 5 Jun 2024 - 31 Jul 2024	0.06 PM / 0.44 PM		0.00 PM / 0.50 PM		-		0.06 PM / 1 PM
Total	0.11 PM / 0.72 PM		0.00 PM / 0.83 PM		0.00 PM / 0.33 PM		**0.11 PM / 2.00 PM**

Figure 30. Person-month allocation and utilisation view.

Salary Budget Resources

	5 JUN 2024 - 30 JUN 2024		1 JUL 2024 - 31 JUL 2024		1 AUG 2024 - 31 AUG 2024		
WORK PACKAGE	USED		USED		USED		TOTAL
WP1 (Admin) 5 Jun 2024 - 31 Aug 2024	50.00 €		0.00 €		0.00 €		50.00 €
WP2 (Communication) 5 Jun 2024 - 31 Jul 2024	50.00 €		0.00 €		-		50.00 €
Total	100.00 €		0.00 €		0.00 €		**100.00 € / 10000.00 €**

Figure 31. Salary allocation and utilisation view

Expenses Budget Resources

	5 JUN 2024 - 30 JUN 2024	1 JUL 2024 - 31 JUL 2024	1 AUG 2024 - 31 AUG 2024	
WORK PACKAGE	USED	USED	USED	TOTAL
WP1 (Admin) 5 Jun 2024 - 31 Aug 2024	0.00 €	0.00 €	0.00 €	0.00 €
WP2 (Communication) 5 Jun 2024 - 31 Jul 2024	200.00 €	0.00 €	-	200.00 €
Total	200.00 €	0.00 €	0.00 €	**200.00 € / 10000.00 €**

Figure 32. Expenses allocation and utilisation view.

5.3.4 Work Packages

The work packages view allows Project Managers to create, view, and update work packages, facilitating detailed management and tracking of various project segments. Figure 33 illustrates an example of the Project Work Packages view, showcasing two work packages - *Admin* and *Communication* - along with their respective periods from June 5, 2024, to July 31, 2024. The figure also displays the personal months and budget allocated to each work package.

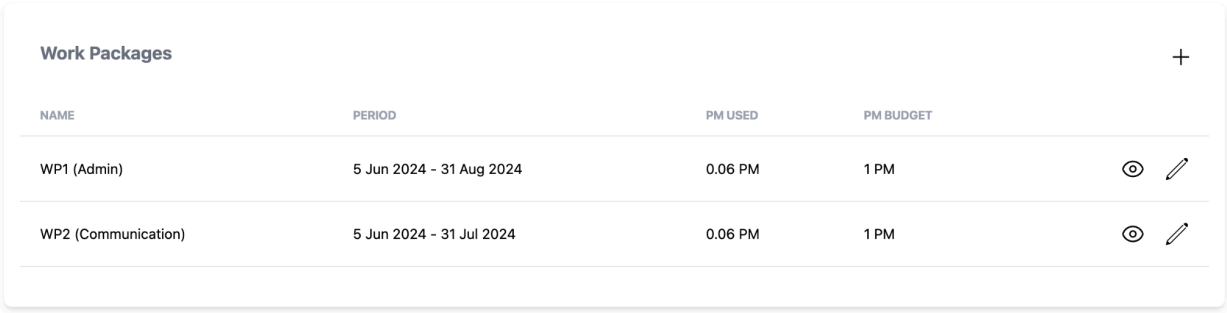


Figure 33. Project work packages view.

5.3.5 Parties and Assigning Parties to Projects

Manage project parties and assign them to specific projects to ensure proper resource allocation and responsibility distribution. Figure 34 displays two parties, *Admin Tartu Linnavalitsus* and *Jaanus Tamm*, assigned to a project with their respective load distribution (20% each) and work hours. Figure 35 illustrates a modal window for adding a party to a project by selecting an employee from the list. Additionally, Figure 36 shows the option to define the distributed load for each employee.

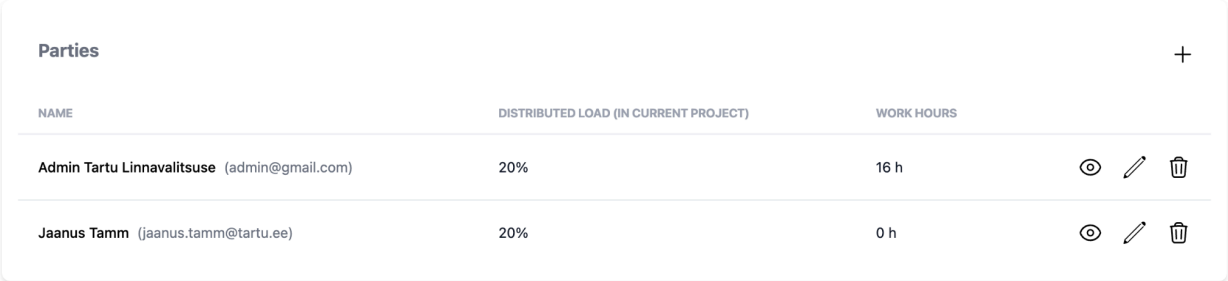


Figure 34. Parties view.

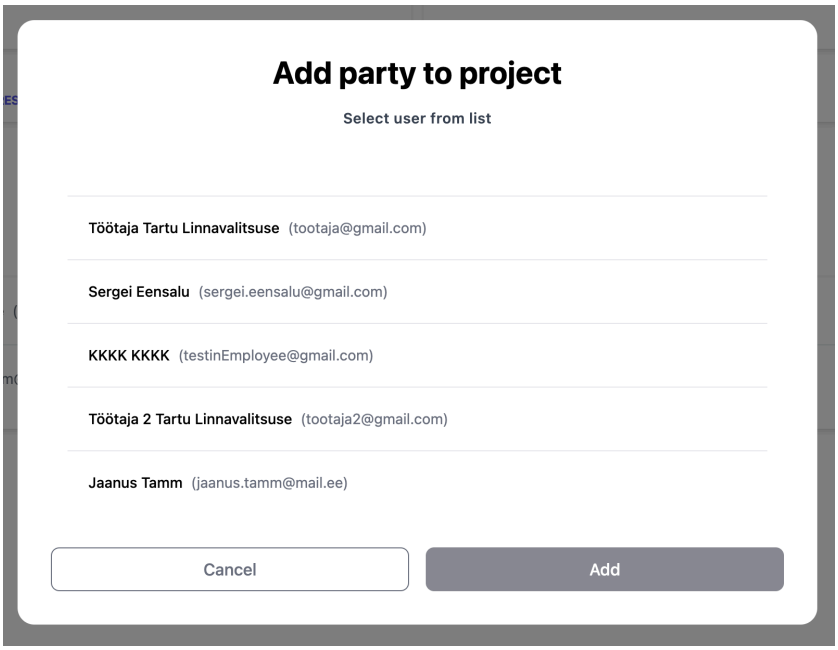


Figure 35. Add a party view.

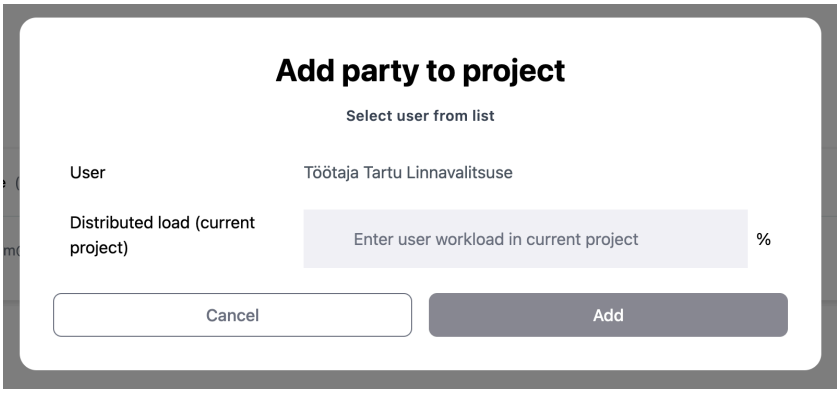


Figure 36. Define party workload view.

5.3.6 Work History

View, update, delete, and create work history entries to ensure accurate records of completed work. Figure 37 illustrates an example of work history entries for two parties, *Admin Tartu Linnavalitsus* and *Jaanus Tamm*. This includes their registered work histories, such as the management history recorded on 05.06.2024, where 8 hours were spent on work package *WP1 (Admin)*.

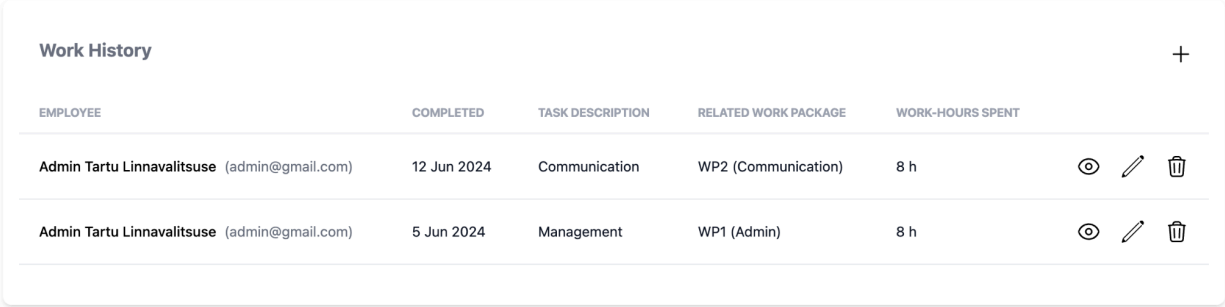


Figure 37. Work histories view.

5.3.7 Expenses

View, create, update, and delete expense records to maintain accurate financial tracking. Figure 38 provides an example of the most recent two expenses added to the work packages *Communication* and *Admin*. Each expense is detailed with a description, type, date, and amount (200 EUR and 45 EUR, respectively).

Generate expense reports grouped by work package to ensure precise financial oversight. Figure 39 illustrates a sample report that groups expenses by work package, showing the total expenses for each work package as well as the overall total for all work packages.

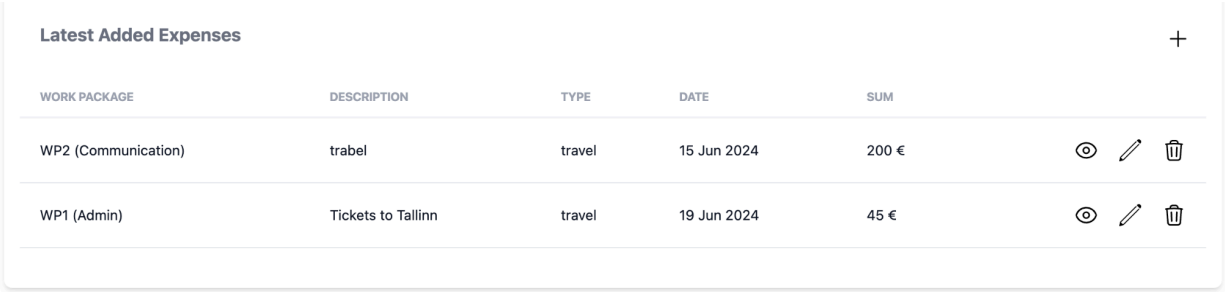


Figure 38. Expenses view.

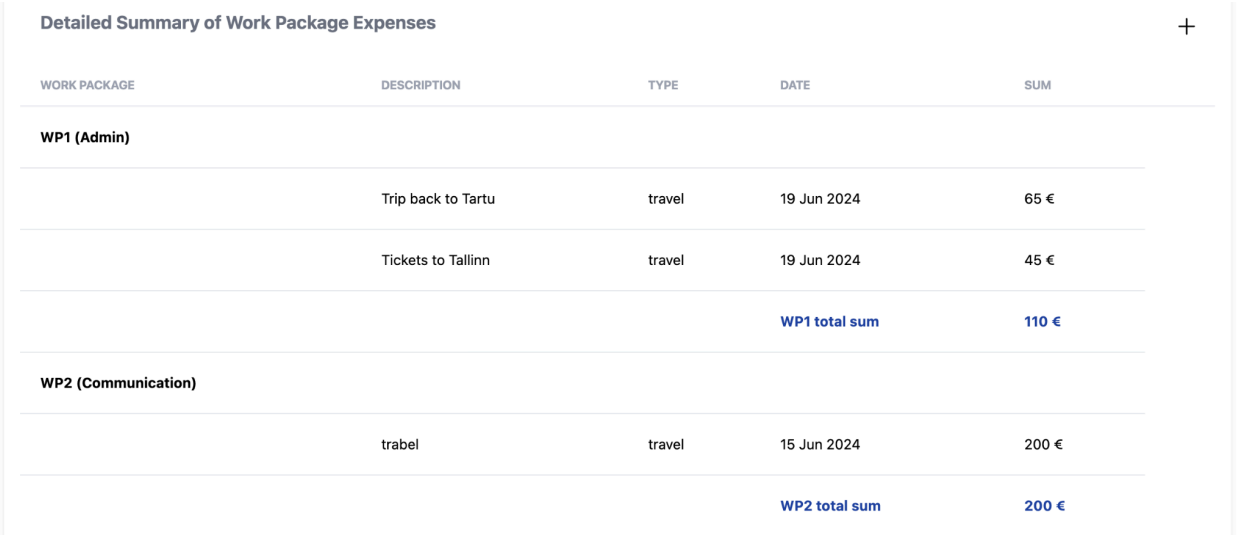


Figure 39. Expenses report view.

6 User Research

The author conducted comprehensive user research of the budget monitoring software to evaluate its functionality, usability, and intuitiveness for users. This evaluation aimed to gather valuable insights into the software's performance and user experience, ensuring it effectively meets user needs and identifying areas for improvement.

6.1 Methodology

The testing process was conducted using Naturalistic Observation methodology, focusing on real-world usage scenarios. Naturalistic Observation is a qualitative research method where researchers study participants' behaviour in their natural settings without intervention [52]. As Cherry K. explains [45], this approach allows for observing how users interact with the system in their natural environment, providing valuable insights into the software's performance and user experience. This method is particularly suitable for this software because it captures authentic user behaviour and identifies potential usability issues that might not surface in a controlled testing environment.

6.2 Testing Setup and Scenarios

The testing will be conducted with two employees of the Tartu City Government, who are the future users of the application. To ensure the purity of the testing, users who have never seen or tested the application were selected. The testing is conducted one-on-one via video call.

To facilitate the testing process, an admin user and an employee user were pre-created by the author and provided to the testing group. The user research consists of five parts: introduction of the testing process, main part of testing: tasks and observation, Net promoter score (NPS) questions, discussion, and conclusion.

In the first part, the testing process will be introduced to the participants. This includes an overview of the testing objectives, procedures, and expectations.

In the second part, participants will perform specific tasks designed to evaluate the functionality, usability, and intuitiveness of the application. These tasks will include creating a new project, adding users, registering work history, adding expenses, setting salaries, and generating reports. Observation points will be noted by the author to assess the ease of use and accuracy of data entry. Sometimes the interviewer might use freely chosen data for the tasks. During testing, participants cannot ask questions from the author, and the author cannot help participants with hints and explanations. Participants will also be asked to verbalise their thoughts and reasoning aloud to understand their thought process while performing tasks.

In the third part of the testing, participants will be asked questions evaluated using the NPS (Net Promoter Score) scale, where they will provide a rating between 0 (not at all likely) and 10 (extremely likely).

In the fourth part of the testing, a free-form discussion will take place where participants can ask questions or discuss their thoughts, suggestions, problems, and general feedback.

The research concludes with thanking the participants for their participation in the testing.

The detailed script for the user research interview can be found in Appendix 3.

6.3 User A results

The author conducted comprehensive user research on July 31, 2024, with an employee from the Tartu City Government to evaluate the functionality, usability, and intuitiveness of the new budget monitoring software. The objective was to gather valuable insights into the software's performance and user experience, ensuring it meets user needs and identifying areas for improvement.

6.3.1 Participant and Session

Position: Analyst

Responsibilities: Management of city property alienation, analysis of rental income, city property purchase and sale transactions, and other related activities within the department.

Date: July 31, 2024

Duration: 32 minutes

6.3.2 Observation

1. Creating a project and work packages: The participant successfully created a project, filling in all necessary details. Initially, the participant attempted to proceed without specifying the expense types, but the software prompted them to complete this requirement, which they did successfully. This step demonstrated the software's ability to guide users through necessary fields. The participant created two new work packages without any issues, indicating the process's intuitiveness.
2. Adding users to a project: Initially, the participant navigated to the user page instead of the project's 'Parties' section. The participant quickly corrected this and added two users to the project. When asked about the meaning of the percentages next to users, the participant correctly identified them as representing workload allocation.
3. Registering work hours: The participant registered work hours quickly and accurately, showing a clear understanding of the process.
4. Adding expenses: Adding expenses was smooth and without complications, highlighting the ease of data entry.
5. Setting salaries: This task presented some challenges. The participant initially looked for salary settings within the project but later understood, with a hint, that salary adjustments are user-specific rather than project-specific. After some searching, the participant successfully located the correct interface for salary adjustments.
6. Generating reports: The participant easily generated a person-month report and a comprehensive work hour report for the current year, indicating the reporting feature's efficiency.

6.3.3 Results

The user testing revealed several key insights:

- The participant found the application relatively intuitive and logical.
- The process of creating projects, work packages, and registering work hours was smooth. However, there was some initial confusion regarding the location of the salary settings.

- The participant suggested that transitioning from Excel or EMDESK to the new software would require a phase where both systems are used simultaneously. This feedback points to a need for clear migration strategies and possibly additional training or support during the transition period.

6.3.4 NPS Evaluation

The participant was asked to rate several aspects of the application using the Net Promoter Score (NPS) scale:

- Intuitiveness: The application was described as "relatively natural and logical."
- Likelihood of use: The participant rated it an 8, indicating a high likelihood of adoption in their work.
- Recommendation: The participant rated it an 8 for recommending the application to colleagues.
- Improvement suggestions: The participant did not have specific suggestions for software features but expressed concerns about transitioning from existing tools to the new software.

6.3.5 Summary

The user research provided valuable feedback, indicating that the application is largely intuitive and meets user needs, particularly in project creation, work package management, and reporting. The main area for improvement is in guiding users more effectively to non-project-specific settings like salary adjustments. The testing confirmed that the software is well-suited for its intended purpose, with high potential for adoption among Tartu City Government employees. Further iterations will focus on refining user guidance and supporting the transition from existing tools to ensure a seamless migration.

6.4 User B results

The author conducted a follow-up user research session on August 5, 2024, with another employee from the Tartu City Government. This session's purpose was to evaluate further the functionality, usability, and intuitiveness of the new budget monitoring software. The

insights gained from this session were crucial in identifying strengths and areas for improvement in the application.

6.4.1 Participant and Session

Position: Mobility specialist
Responsibilities: Evaluation of mobility solutions in plans and projects, commissioning studies, and drafting mobility-related development documents.
Date: August 5, 2024
Duration: 37 minutes

6.4.2 Observation

1. Creating a project and work packages: The participant successfully created a project and corresponding work packages. Initially, there was some uncertainty about the location of the "Create Project" button, but this was quickly resolved by exploring the interface. The participant expressed confusion about the purpose of the budget form and the future use of expense types.
2. Adding users to a project: The participant added two users to the project without issues, assigning appropriate workload percentages. This task was completed smoothly.
3. Registering work hours: The participant efficiently registered work hours but inquired where to find detailed descriptions of completed work. A suggestion was made to allow registering tasks over a range of days instead of day by day for convenience.
4. Adding expenses: The participant initially revisited the project to add new expense types before adding expenses, demonstrating an understanding of the system's requirements. This task was completed successfully.
5. Setting salaries: The participant found the appropriate location to add salaries easily and understood why salary adjustments were user-specific rather than project-specific.

6. Generating reports: The participant located the budget graph showing person-month usage, correctly interpreted the information, and generated a report for a worker's hours for the current year without any complications.

6.4.3 Results

The user testing revealed several key insights:

- The software effectively guided the participant through necessary fields, such as expense types, ensuring all required information was entered.
- There was a brief confusion regarding the navigation to the correct sections, but the participant quickly adapted and found the required functionalities.
- The participant appreciated the ease of entering data but suggested improvements for registering work over multiple days and generating detailed reports.

6.4.4 NPS Evaluation

- Intuitiveness: The participant found the application relatively intuitive but noted some areas required more thought, particularly in creating budget categories.
- Likelihood of use: Rated 5, indicating a moderate likelihood of adoption.
- Recommendation: Rated 5, suggesting moderate enthusiasm for recommending the application to colleagues.
- Improvement suggestions:
    - Improve calendar functionality to prevent jumping when changing months.
    - Allow for more customizable budget categories.
    - Enhance report tables to display forecasts or trends.
    - Facilitate the generation of various reports in different formats, such as Excel.

6.4.5 Summary

This user research session highlighted the application's strengths in data entry and intuitive project management. However, it also identified areas for improvement, such as smoother calendar navigation, customizable budget categories, and enhanced reporting features.

These insights are invaluable for refining the application to better meet the needs of Tartu City Government employees, ensuring a more user-friendly experience.

6.5 Conclusion

The user research sessions with Tartu City Government employees provided crucial insights into the new budget monitoring software's usability and effectiveness. The software demonstrated strengths in guiding users through necessary fields, ensuring accurate data entry, and supporting project management tasks efficiently. Participants found the application intuitive in creating projects, work packages, and registering work hours.

The Net Promoter Score (NPS) evaluations indicated a mixed likelihood of adoption and recommendation, with scores of 8 and 5, reflecting the application's potential with further refinements. The Naturalistic Observation approach effectively captured real-world user interactions, providing insights that might not surface in controlled environments. This feedback will be instrumental in refining the system to meet user needs better and ensure a smooth transition to the new software.

7 Possibilities for Further Development

In the next iteration of the budget monitoring software, several significant enhancements and improvements are planned to optimise functionality and user experience further. A key focus will be the transition from the current NoSQL database to a relational SQL database, which will enhance data management, querying capabilities, and scalability. This migration will allow for more complex queries and improved data integrity, which are crucial for handling large datasets and generating detailed reports.

Enhanced reporting features are also on the agenda, with plans to implement more detailed and customizable reports. These reports will provide deeper insights into project performance and resource utilisation, helping managers make more informed decisions. Features such as trend analysis, forecast reports, and real-time data updates will be valuable additions.

User feedback is invaluable, and incorporating it into the development process will address current pain points and enhance usability. Planned upgrades based on feedback include improving the user interface to make it more intuitive and user-friendly. Additionally, transitioning from Excel or EMDESK to the new software would benefit from a phase where both systems are used simultaneously. This points to the potential benefit and need for future data importing functionality, ensuring a smooth transition and data migration.

Initially, the project scope was limited to a web version for desktop usage, and the extension to include mobile accessibility was not part of the initial plan. Modern design principles and responsive layouts will ensure better accessibility across devices, including smartphones.

Furthermore, the budget calculation and person-month logic will be upgraded to include Estonian holidays. This enhancement involves integrating a mechanism to account for non-working days due to national holidays in Estonia. Initially, the development team explored existing services and APIs that provide holiday data. However, finding a service that seamlessly integrated with the software and offered comprehensive coverage for all Estonian holidays proved challenging. As a result, implementing a custom solution for managing holiday data and incorporating it into the budget calculations was deemed necessary but was postponed due to scope constraints. This upgrade will ensure more

accurate planning and allocation of resources by adjusting schedules and budgets accordingly to reflect non-working days.

These improvements will ensure the software meets the evolving needs of the Tartu City Government, ensuring its long-term success and adaptability. By focusing on scalability, user experience, integration, and accurate resource planning, the software will be well-equipped to handle future challenges and opportunities.

8 Summary

This thesis aimed to develop and implement a robust, scalable, and user-friendly web-based budget monitoring system specifically tailored to the needs of the Tartu City Government. The transition from a traditional Excel-based approach to an integrated web-based solution was essential to address significant limitations in data integrity, scalability, and user collaboration.

The developed system successfully leverages modern web technologies, including Vue.js, Tailwind CSS, Google Cloud Platform, and Firebase, to provide enhanced data management and reporting capabilities. By implementing a centralised database and robust backend, the software ensures data synchronisation, improved querying capabilities, and better scalability. As a result, the Tartu City Government now has a more reliable and efficient tool for budget monitoring and management.

One of the critical achievements of this project is the significant improvement in user experience. The intuitive user interface, combined with streamlined data entry processes and updates, has made the system easier to use, reducing the likelihood of errors and enhancing overall efficiency. The user roles and permissions feature ensures that data access is appropriately restricted, maintaining data security while allowing for effective collaboration.

Additionally, the new system's enhanced reporting features provide deeper insights into project performance and resource utilisation, enabling more informed decision-making. The ability to generate detailed and customizable reports helps project managers and other stakeholders monitor progress and manage resources more effectively.

Incorporating user feedback throughout the development process was crucial in identifying pain points and areas for improvement. The iterative development approach ensured that the final product met the users' needs and expectations, resulting in a more refined and functional system.

The thesis also identified several areas for future development to further enhance the system's capabilities. These include migrating to a relational SQL database for even better data management and querying, developing a comprehensive backend, incorporating advanced reporting features, and continuously improving the user interface based on

ongoing user feedback. Additionally, integrating Estonian holidays into the budget calculation and person-month logic will ensure more accurate planning and resource allocation.

In conclusion, the development and implementation of this web-based budget monitoring system addresses the critical limitations of the previous Excel-based approach, providing a more scalable, accurate, and user-friendly solution. This new system enhances the efficiency of budget management and reporting and sets the stage for future improvements and adaptations. The author of this work will continue to develop the software for the Tartu City Government as a freelancer, ensuring its long-term success and adaptability to meet the evolving needs of the city government. Through the integration of modern technologies and continuous user-centred development, the Tartu City Government is now better equipped to manage its financial resources and ensure transparency and accountability in its operations.

References

[1] Larson, E. W., & Gray, C. F. (2017). “Project Management: The Managerial Process (7th ed.)”. McGraw Hill. Chapter 1, “Defining the Project”, pp. 2-6.

[2] Few, S. (2013). “Information Dashboard Design: Displaying Data for At-a-Glance Monitoring.”. Analytics Press. Chapter 1, pp. 3-7.

[3] Kerzner, H. (2017). “Project Management: A Systems Approach to Planning, Scheduling, and Controlling (12th ed.)”. John Wiley & Sons Inc. Chapter 11, Section 11.11, “Work Breakdown Structure”, pp. 434-437.

[4] Horngren, C. T., Datar, S. M., & Rajan, M. V. (2015). “Cost Accounting: A Managerial Emphasis (15th ed.)”. Pearson. Chapter 2, "An Introduction to Cost Terms and Purposes," pp. 42-60.

[5] Kerzner, H. (2017). “Project Management: A Systems Approach to Planning, Scheduling, and Controlling (12th ed.)”. John Wiley & Sons Inc. Chapter 1, Section 1.4, “Defining the Project Manager’s” Role, pp. 12-15.

[6] Horngren, C. T., Datar, S. M., & Rajan, M. V. (2015). "Cost Accounting: A Managerial Emphasis (15th ed.).". Pearson. Chapter 7, "Flexible Budgets, Direct-Cost Variances, and Management Control," pp. 234-256.

[7] Larson, E. W., & Gray, C. F. (2017). “Project Management: The Managerial Process (7th ed.)”. McGraw Hill. Chapter 5, “Estimating Project Times and Costs”, pp. 130-132.

[8] Baye, M. R. (2010). “Managerial Economics and Business Strategy (7th ed.)”. McGraw-Hill/Irwin. Chapter 5, "The Production Process and Costs" pp. 155-182.

[9] Sommerville, I. (2011). “Software Engineering (9th ed.)”. Pearson. Chapter 16, "Software Reuse" pp. 426-440.

[10] Sommerville, I. (2011). Software Engineering (9th ed.). Pearson. Chapter 18, "MobileApp Design," pp. 403-407.

[11] Google. (2024). "Google Cloud Products". Retrieved on 20 June 2024. [Online]. Available: https://cloud.google.com/products.

[12] European Union. “The European Union”. Retrieved on 20 June 2024. [Online]. Available: https://european-union.europa.eu/easy-read_en.

[13] Voigt, P., & Von dem Bussche, A. (2017). "The EU General Data Protection Regulation (GDPR): A Practical Guide". Springer. Chapter 1, "Introduction and 'Checklist'," pp. 1-30.

[14] EMDESK. (2024). "Your Comprehensive Horizon Europe Software Solution". Retrieved on 20 June 2024. [Online]. Available: https://www.emdesk.com

[15] Schiavo-Campo, S., & Tommasi, D. (1999). "Managing Government Expenditure". Asian Development Bank. Chapter 6, "Assuring Compliance in Budget Execution," pp. 1-6.

[16] Diamond, J., & Khemani, P. (2005). "Introducing Financial Management Information Systems in Developing Countries". International Monetary Fund. Chapter 1, pp. 3-4.

[17] Weikart, L. A., Chen, G. G., & Sermier, E. (2013). "Budgeting and Financial Management for Nonprofit Organizations". CQ Press. Chapter 3, pp. 87-94.

[18] Phocas Software. (n.d.). "Excel-based vs web-based budgeting and forecasting". Retrieved on 20 June 2024. [Online]. Available: https://www.phocassoftware.com/resources/blog/budgeting-software-vs-excel.

[19] Benninga, S. (2014). "Financial Modeling (4th ed.)". MIT press. Chapters 33 and 35, pp. 855-932.

[20] M. Glinz, "On Non-Functional Requirements," 15th IEEE International Requirements Engineering Conference (RE 2007), Delhi, India, 2007, pp. 21-26, doi: 10.1109/RE.2007.45.

[21] Vue.js. (n.d.). "What is Vue?". Retrieved on 20 June 2024. [Online]. Available: https://vuejs.org/guide/introduction.html#what-is-vue

[22] PixelCrayons. (n.d.). "7 Compelling Reasons Why VueJS is the Perfect Framework for Modern Businesses". Written by Varun Bhagat. Retrieved on 20 June 2024. [Online]. Available: https://www.pixelcrayons.com/blog/dedicated-teams/why-use-vuejs/

[23] Bytesfarms. (n.d.). "Vue Vs React: Which Is The Better Front End Js Framework." Retrieved on 21 July 2024. [Online]. Available: https://bytesfarms.com/vue-vs-react-which-is-the-better-front-end-js-framework/

[24] Zero To Mastery. (n.d.). "Angular vs React vs Vue: The Best Framework for 2024." Retrieved on 21 July 2024. [Online]. Available: https://zerotomastery.io/blog/angular-vs-react-vs-vue/

[25] Amazon Web Services. "What is AWS?". Retrieved on 21 July 2024. [Online]. Available: https://aws.amazon.com/what-is-aws/

[26] Banks, A., & Porcello, E. (2016). Learning React: Functional Web Development with React and Redux. O'Reilly Media. Chapter 1, pp. 1-25.

[27] Seshadri, S., & Green, B. (2018). Angular: Up and Running. O'Reilly Media. Chapter 1, pp. 3-22.

[28] Tailwind CSS. (n.d.). "Tailwind CSS: A Utility-First CSS Framework for Rapidly Building Custom Designs." Version 1.9.0. Retrieved on 21 July 2024. [Online]. Available: https://v1.tailwindcss.com/

[29] Sosinsky, B. (2011). Cloud Computing Bible. Wiley. Chapter 9, Using Amazon Web Services, pp. 180-184.

[30] Firebase. (n.d.). "SDKs and Client Libraries." Retrieved on 21 July 2024. [Online]. Available: https://firebase.google.com/docs/firestore/client/libraries

[31] Richman, J. (2023, April 12). "Firestore Query & Record Limitations: How to Work Around It." Estuary. Retrieved on July 22, 2024. [Online]. Available: https://estuary.dev/firestore-limitations/

[32] Firebase. (2024, July 19). "Understand Cloud Firestore Billing." Retrieved on July 22, 2024. [Online]. Available: https://firebase.google.com/docs/firestore/pricing

[33] Google Cloud. (n.d.). "What is NoSQL?" Retrieved on July 22, 2024. [Online]. Available: https://cloud.google.com/discover/what-is-nosql

[34] Kareem, M. (2023, September 6). "5 reasons why you should reconsider using Firebase." Medium. Retrieved on July 22, 2024. [Online]. Available: https://mohammedkareem.medium.com/5-reasons-why-you-should-reconsider-using-firebase-6b167496906f

[35] Firebase. (n.d.). “Pricing”. Firebase. Retrieved on July 22, 2024. [Online]. Available: https://firebase.google.com/pricing

[36] Oracle. (2024). “What is MySQL?”. Retrieved on July 22, 2024. [Online]. Available: https://www.oracle.com/mysql/what-is-mysql/

[37] PostgreSQL Global Development Group. (2024). “What is PostgreSQL?”. Retrieved on July 22, 2024. [Online]. Available: https://www.postgresql.org/about/

[38] Firebase. (2024). “Firebase Security Rules”. Retrieved on July 22, 2024. [Online]. Available: https://firebase.google.com/docs/rules

[39] Firebase. (2024). “Basic Security Rules”. Retrieved on July 22, 2024. [Online]. Available: https://firebase.google.com/docs/rules/basics

[40] Firebase. (2024). “Firebase Hosting”. Retrieved on July 22, 2024. [Online]. Available: https://firebase.google.com/docs/hosting

[41] Firebase. (2024). “Firebase Authentication”. Retrieved on July 22, 2024. [Online]. Available: https://firebase.google.com/docs/auth

[42] Firebase. (2024). “Manage User Sessions”. Retrieved on July 22, 2024. [Online]. Available: https://firebase.google.com/docs/auth/admin/manage-sessions

[43] Wolford, B. (2024). "What is GDPR, the EU’s new data protection law?" GDPR.eu. Retrieved on July 22, 2024. [Online]. Available: https://gdpr.eu/what-is-gdpr/

[44] Firebase. (2024). "Firestore multi-region locations". Retrieved on July 22, 2024. [Online]. Available: https://firebase.google.com/docs/firestore/locations#location-mr

[45] Cherry, K. (2023, December 1). "What is naturalistic observation?" Verywell Mind. Retrieved on July 22, 2024. [Online]. Available: https://www.verywellmind.com/what-is-naturalistic-observation-2795391

[46] Sosinsky, B. (2011). Cloud Computing Bible. Wiley. Chapter 8, Using Google Web Services, pp. 153-155.

[47] Google Cloud. (n.d.). "What is GCP?" Retrieved on 20 June 2024. [Online]. Available: https://cloud.google.com/gcp

[48] DuBois, P. (2013). MySQL (5th ed.). Addison Wesley. Introduction, pp. 1 - 4.

[49] Momjian, B. (2000). PostgreSQL: Introduction and Concepts. Addison-Wesley. Chapter 1, pp. 1- 4.

[50] Shim, J. K., & Siegel, J. G. (2008). "Budgeting Basics and Beyond" (2nd ed.). John Wiley & Sons. Chapter 19, "Software Packages: Computer-based Models and Spreadsheet Software," pp. 279-290.

[51] David, M., & Yi, M. (n.d.). "What is Net Promoter Score (NPS) and When to Use It" Chartio. Retrieved on 26 July 2024. [Online]. Available: https://chartio.com/learn/product-analytics/what-is-net-promoter-score-nps/

[52] Bhandari, P. (2023, June 22). Naturalistic Observation | Definition, Guide, & Examples. Scribbr. Retrieved August 5, 202. [Online]. Available: https://www.scribbr.com/methodology/naturalistic-observation/

Appendix

I. Source Code

Budget Monitoring Web Repository

The web application developed as part of this master's thesis is contained within a GitLab repository.

1. Budget Monitoring Web Repository
   This repository contains the source code for the web application, built using the Vue.js JavaScript framework.

Access and Privacy

The repository is private. The decision to keep the repository private is to protect sensitive information related to the Tartu City Government's internal processes and to maintain security over the implementation details. Additionally, the code may contain proprietary elements developed specifically for this project, which are not intended for public distribution.

To access the code, contact the author:

Sergei Eensalu
Email: sergei.eensalu@gmail.com

II. Demo Environment

The demo environment for the web application can be accessed at the following URL.

1. Budget Monitoring Demo Environment

   This demo environment showcases the functionalities of the Budget Monitoring Web Application, including project management, work hour tracking, expense management, and reporting tools. Users can explore the interface and features designed to streamline the budget monitoring process for EU projects.

The demo environment access is private. To access the environment, contact the author:

Sergei Eensalu
Email: sergei.eensalu@gmail.com

III. Script of User Research Interview

This script is prepared for all participants testing the budget monitoring software using the User Research observation method.

1. Introduction
    1.1. Author Introduction
    1.2. Introduction to User Research and Guidelines
    1.3. Permission to Record Video Call
    1.4. Thanking for Participation
2. Testing Flow
    2.1. Project and work package creation
    - Task: Create a new project (start date 01.03.2024) and two associated work packages (start date 01.03.2024).
    - Observation Points: Assess the intuitiveness of the process and ease of adding details.

    2.2. Adding party to the project
    - Task: Add yourself to the project and one additional user.
    - Observation Points: Verify the clarity of the process and successful linkage to the project.

    2.3. Work history registration
    - Task: Register work history for yourself and another user. For yourself, choose the first work package (wp1); for the other user, choose the second work package (wp2).
    - Observation Points: Evaluate the ease of entry, accuracy, and linkage to projects and work packages.

    2.4. Adding expenses to project
    - Task: Add at least two different expenses to a project.

- Observation Points: Check if the interface is straightforward and if expense details are correctly saved. Ensure the user understands they need to return to the project.

2.5. Setting new party salary

- Task: Set a new salary for a party from a specific start date (e.g., 01.01.2024).
- Observation Points: Ensure salary details are accurately recorded and displayed, and that the user understands they need to exit the project and go to user settings.

2.6. Finding project person-month report

- Task: Locate the person-month report for a project.
- Observation Points: Assess the report's accuracy, comprehensiveness, and accessibility. Ensure the user understands they need to return to the project view to find the report and clarify their understanding of the report.

2.7. Generating user work hours report for current year

- Task: Generate a user work hours report for the current year.
- Observation Points: Confirm the report's accuracy, comprehensiveness, and ease of generation.

3. General questions about the new application
   - 3.1. How intuitive did you find using the new application?
   - 3.2. How likely is it that the new application would be used in your work?
   - 3.3. How likely are you to recommend the new application to your colleagues?
   - 3.4. Do you have any ideas or suggestions for improvement?
4. Free form discussion
   - Allow participants to discuss any additional thoughts, experiences, or feedback related to the application.
5. Conclusion
   - Thank the participants once again for their participation in the testing.

IV. Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Sergei Eensalu

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Budget Monitoring Web Application For Managing EU Projects”, supervised by Pelle Jakovits and Gert Kanter

    1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

    1.2. to be published via the web of Tallinn University of Technology, incl. To be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2. I am aware that the author also retains the rights specified in clause 1 of the nonexclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons’ intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

Sergei Eensalu
09.11.2024

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period