TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technologies

Kreete Keres 232867IABB, Demi Heinsoo 233141IABB

# UX/UI design and development of Buka: a workforce management solution for service industry companies

Bachelor's thesis

Juhendajad: Karl-Erik Karu MSc,
Viljam Puusep MSc

Tallinn 2026

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Kreete Keres 232867IABB, Demi Heinsoo 233141IABB

# Buka UX/UI disain ja arendus: tööjõu- ja graafikuhalduse lahendus teenindussektori ettevõtetele

Bakalaureusetöö

Juhendajad: Karl-Erik Karu MSc,
Viljam Puusep MSc

Tallinn 2026

## 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: Kreete Keres

[08.05.2026]

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: Demi Heinsoo

[09.05.2026]

# Abstract

Workforce scheduling in the service industry remains one of the few business-critical processes still largely handled through manual and fragmented methods, leading to inefficiency, communication breakdowns, and a fundamental lack of transparency that affects both employees and employers. This thesis explores the UX/UI design and development of Buka, a workforce management application for service industry companies, aimed at simplifying scheduling, improving communication, and increasing visibility into schedule and wage management.

The development process combined competitor analysis, qualitative research methods, UX/UI design principles, and iterative Scrum-based software development practices. UX interviews and usability testing sessions were conducted with employees, managers, and a professional UX specialist to validate the concept and improve the application.

The final solution includes automatic schedule generation, real-time wage calculation, shift swapping, open shifts, notifications, and the submission of vacation, sick leave, and day-off requests, with particular attention to usability, accessibility, and automation. The results demonstrate that modern workforce management solutions can reduce manual administrative work, improve communication, and increase efficiency within service industry companies.

This thesis is written in English and consists of 67 pages, including 7 chapters, 47 figures, and 2 tables.

# Lühikokkuvõte

**Buka UX/UI disain ja arendus: tööjõu- ja graafikuhalduse lahendus teenindussektori ettevõtetele.**

Teenindussektori tööjõu graafikuhaldus on jäänud üheks väheseks ärikriitiliseks protsessiks, mida hallatakse endiselt suuresti käsitsi ja killustatud meetoditega. Ebaefektiivsus, suhtlushäired ning läbipaistvuse puudumine mõjutavad nii töötajaid kui tööandjaid. Käesolev lõputöö käsitleb teenindussektori ettevõtetele mõeldud tööjõuhalduslahenduse Buka UX/UI disaini ja arendust. Töö eesmärk oli luua kaasaegne ja kasutajasõbralik rakendus, mis lihtsustab töögraafikute koostamist, parandab töötajate ja juhtide vahelist suhtlust ning tagab suurema läbipaistvuse nii graafiku- kui palgahalduses.

Arendusprotsessis rakendati konkurentsianalüüsi, kvalitatiivseid uurimismeetodeid, UX/UI disainipõhimõtteid ning iteratiivseid tarkvaraarendustavasid. Kontseptsiooni valideerimiseks ja kasutusmugavuse parandamiseks viidi läbi mitmeid UX-intervjuusid ning kasutajatestimise seansse töötajate, juhtide ja kogenud UX-spetsialistiga.

Valminud lahendus hõlmab funktsionaalsusi nagu automaatne graafikute genereerimine, reaalajas palgaarvestus, vahetuste vahetus, teavitussüsteem ning puhkuste, haiguslehtede ja vabade päevade taotlemine. Arenduse käigus pöörati erilist tähelepanu kasutusmugavusele, ligipääsetavusele, automatiseerimisele ning intuitiivse kasutajakogemuse kujundamisele.

Töö tulemused kinnitavad, et kaasaegsed tööjõuhalduslahendused suudavad märkimisväärselt vähendada käsitsi tehtavat haldustööd, parandada meeskonnasisest suhtlust ning tõsta teenindussektori ettevõtete tööefektiivsust.

Lõputöö on kirjutatud inglise keeles ning sisaldab 67 lehekülge, 7 peatükki, 47 joonist ja 2 tabelit.

# List of abbreviations and terms

| | |
|---|---|
| AI | Artificial Intelligence |
| GDPR | General Data Protection Regulation |
| Hi-fi | High-fidelity |
| Lo-fi | Low-fidelity |
| SME | Small and Medium-sized Enterprise |
| TalTech | Tallinn University of Technology |
| UI | User Interface |
| UX | User Experience |

# Table of Contents

# List of figures

# List of tables

# 1 Introduction

Shift scheduling remains one of the few business-critical processes that has not been fully digitised. Although most administrative operations have moved online, many cafés, restaurants, and service-sector businesses still depend on spreadsheets, paper schedules, and informal messaging channels to organise and manage employee shifts. This thesis presents the UX/UI design and development of Buka, a workforce management application built to address that gap. The following sections describe the background and problem the project addresses, the goals of the thesis, and the structure of the chapters that follow.

## 1.1 Background and problem

Workforce scheduling in the service industry is broken. Despite the wide availability of modern digital tools, many service and seasonal workplaces still rely on spreadsheets, paper-based solutions, manager notes, and informal messaging platforms. These methods are flexible, but incompatible with modern workforce management, creating unnecessary burdens for both employees and employers.

At the heart of the problem lies the absence of a centralised, real-time system. Employees have no immediate visibility into upcoming shifts, leave request statuses, or schedule changes. Every modification depends on manual intervention by a manager, creating an organisational bottleneck that is slow, error-prone, and entirely dependent on one person's availability.

The consequences are both operational and human. Manual scheduling can lead to uncovered shifts, inaccurate tracking of working hours, and salary mistakes, while limited transparency makes it difficult for employees to organise their personal lives and creates additional stress. Scheduling problems are also among the primary reasons young workers leave their jobs. The "Stable Scheduling Study" [1] proved this concretely: consistent schedules and digital shift control reduced turnover, raised productivity by 5%, and increased median sales by 7%. The problem is particularly relevant for younger

employees aged 18 to 25, students, and part-time employees, whose working days are irregular. The Deloitte Global Gen Z and Millennial Survey [2] supports this trend, identifying work-life balance and "digital maturity" as top priorities for younger generations.

Wage transparency adds another important dimension. In Estonia, payroll calculations involve multiple rules and regulations, including regular working hours, overtime and undertime, night shift and public holiday bonuses, as well as vacation pay, sick leave, and study leave, each calculated differently [3]. Without a system that tracks these components, employees, especially first-time workers, cannot understand how their salary is formed, reducing trust and creating constant uncertainty.

The existing system no longer meets the expectations of today's digital-native workforce. Schedule visibility, leave management, shift swapping, and wage transparency all have clear digital solutions, yet these challenges remain largely unresolved in the service sector despite their significant economic and social impact. A user-centred scheduling system is well-positioned to address these issues.

## 1.2 Goal

The goal of this thesis is to research, design, and validate a user-centred digital workforce management solution for the Estonian service sector. The primary aim is to transition from fragmented, error-prone manual scheduling practices to a structured digital environment that addresses the needs of both employees and employers.

The first major objective is to improve workplace transparency and give employees greater control over their schedules. The research aims to explore how a digital system can provide employees with independent, real-time access to their work-related data, such as schedule predictability and leave management. A key focus is on reducing the lack of transparency in the sector, especially regarding how wages are calculated under Estonian labor regulations and helping younger employees better understand their earnings and employment rights.

The second objective is to improve operational accuracy and reduce administrative workload. This includes exploring how automation can reduce human errors in tracking

overtime, leave balances, and shift changes. The thesis also examines how digital workflows can replace informal communication methods and help ensure that scheduling follows legal and organisational requirements with less manual work.

Finally, the research addresses the specific needs of SMEs within the local context. Although global workforce management tools already exist, this study focuses on creating a solution that is simple, affordable, and suitable for Estonia's legal and regulatory requirements. The aim is to offer an alternative to both existing enterprise software and the manual tools still widely used in the service-industry.

## 1.3 Thesis structure

The thesis is structured into seven chapters, each focusing on a different stage of the research, design, development, and evaluation process of the Buka workforce management application.

Chapter 1 introduces the background, problem, and goals of the thesis. It explains the challenges related to workforce scheduling and communication within the service industry and highlights the need for a modern, user-friendly workforce management solution. The goals of the thesis are defined, followed by an overview of the thesis structure.

Chapter 2 describes the methodology used throughout the project. It explains the qualitative research methods applied to gather user feedback and validate the application concept, as well as how Scrum methodology, UX/UI design principles, and design thinking practices were implemented during the development process. In addition, the chapter presents the development and auxiliary tools used throughout the project.

Chapter 3 focuses on the requirements analysis of the application. It begins with an overview and comparison of existing workforce management solutions available on the market, including a comparative analysis of their features and limitations. Based on competitor analysis and user feedback, the chapter then identifies the key insights and defines the business, functional, supporting, and non-functional requirements for the Buka application.

Chapter 4 describes the UX/UI design and development process of the application. It covers the creation of user personas, the development of lo-fi and hi-fi prototypes, visual design decisions, and the iterative testing and validation process conducted with target users and a professional UX designer. The chapter also explains how feedback gathered from interviews influenced the improvement and refinement of the final solution.

Chapter 5 presents the development and main functionalities of the application. It describes the goals and scope of the solution and provides an overview of the implemented features. The chapter also explains the practical workflow and everyday use of the final application.

Chapter 6 analyses the overall outcomes of the project. It includes an overall assessment of the application, teamwork analysis, market positioning and competitive evaluation, business plan, lessons learned during the development process, and future development possibilities for the application.

Chapter 7 provides a summary of the thesis by presenting the key findings, conclusions, and outcomes of the project.

# 2 Methodology

This chapter describes the methodological foundation of the thesis. It begins by introducing Buka as a startup case and clarifying the scope of this thesis within the broader group project. It then outlines the qualitative research methods used to understand user needs and inform design decisions. The chapter continues with a detailed overview of all tools applied throughout the project lifecycle, covering both development and auxiliary tools. Finally, it explains how Scrum methodology, UX/UI design principles, and design thinking practices were combined to support an iterative and user-centred development process.

## 2.1 Case description

This thesis was developed within the context of Buka, a startup project focused on creating an automated workforce management and shift scheduling solution. The system combines automated schedule generation with transparent wage calculation based on Estonian payroll regulations to provide greater clarity and efficiency for both employees and employers.

The project is carried out as a collaborative effort divided into two parallel theses, each focusing on a different part of the system. This thesis focuses on the business aspects of the product and the frontend development of the application. It includes competitor analysis, user research, requirement analysis, and the definition of the application's business value and market positioning. In addition, the thesis is responsible for the user-centred aspects of the system, including UX/UI design and frontend development, with the aim of creating an intuitive solution that meets the practical needs of the target audience.

In contrast, the companion thesis focuses on the internal system logic and technical infrastructure. It addresses the back-end development, including the design of automated scheduling algorithms, the mechanics of shift-swapping and other core functionalities, and the complex calculation engines required for payroll and workforce management.

Buka is designed to generate optimized work schedules by considering employee availability, organisational rules, and both hard and soft constraints, such as employees' expected monthly working hours. The system aims to improve efficiency for both employees and employers while maintaining a strong focus on usability and user experience. By reducing the need for manual scheduling and adjustments, Buka saves time and helps organisations respond more effectively to changes in schedules and workforce availability.

## 2.2 Qualitative research methods

This thesis follows a qualitative research approach to better understand how service sector employees and managers experience work schedule management in practice. Instead of focusing on large-scale numerical data, the research aimed to explore users' frustrations, expectations, workflows, and coping mechanisms related to scheduling and workplace communication. This approach was chosen because the problem is fundamentally human-centred, involving stress, transparency, communication, and trust, which cannot be fully understood through quantitative methods alone. As emphasized by Creswell (2018) [4], qualitative research is particularly suitable when the objective is to understand the meanings individuals assign to their experiences and the context in which these experiences occur.

To support this approach, UX interviews and usability testing were conducted throughout all stages of the development process, including lo-fi and hi-fi prototypes, and the final application implementation. UX interviews are a widely used user research method in UX design, that helps designers better understand users' needs, behaviours, expectations, and pain points through structured or semi-structured conversations [5]. In the context of Buka, the testing sessions were conducted remotely via Microsoft Teams [6], allowing participants to interact with the prototypes and final application in their natural environments while sharing their screens for real-time observation. A more detailed overview of the testing and development process is presented in Chapter 4.4.

The iterative testing process ensured that user feedback was continuously integrated into the development cycle, enabling usability issues and workflow inefficiencies to be identified and addressed at an early stage. Early lo-fi testing focused on validating core

concepts, information architecture, and user flows, while hi-fi prototype testing concentrated on interaction design, visual clarity, and task efficiency. Final application testing was used to evaluate the usability and functionality of the fully implemented system.

The success of the user-centred design approach was evaluated through usability testing with the target users [7], focusing on qualitative indicators such as task success, perceived ease of use, intuitiveness, and overall user satisfaction. The overall objective was to demonstrate that a scheduling system developed through direct user involvement can improve transparency and trust, reduce administrative burden for managers, and support a healthier work-life balance for employees.

## 2.3 Tools

The development of Buka involved the use of a wide range of tools throughout the entire product lifecycle, including ideation, user interface design, software development, user research, and market analysis. These tools were organised into two main categories: development tools and auxiliary tools. The following section provides a detailed overview of each tool, its purpose, and how it was applied to support the previously mentioned tasks.

### 2.3.1 Development tools

**Visual Studio Code (VS Code)** [8] - Visual Studio Code was selected due to the team's prior experience with the tool during their studies. It is a highly flexible development environment that offers integrated support for debugging, version control, and an extensive library of extensions. Additionally, it supports the integration of various AI-based tools that enhance the development process. VS Code served as the primary environment for developing both the front - and backend of Buka.

**Gitlab** [9] - GitLab was chosen based on the team's familiarity with the platform, as well as its availability through student licensing for TalTech students. It functioned as the main code repository and version control system, enabling efficient branch management, issue tracking, and collaboration among team members. These features supported a structured and smooth development workflow.

### 2.3.2 Auxiliary tools

**Figma** [10] - Figma was selected due to its popularity in UI design and prototyping. In this project, it was used to create lo-fi prototypes, which supported early-stage ideation, user testing, and user research. The main purpose was making the design of initial layouts for both the company manager and employee views of each page before transitioning to hi-fi prototypes.

**Lovable** [11] - Lovable acted as an AI-assisted tool for generating frontend interfaces. It was primarily used to transform Figma designs into functional, high-quality hi-fi prototypes. Lovable supported iterative user testing by enabling quick modifications to prototypes based on user feedback, before the actual development phase.

**Google Drive** [12] - Google Drive was used to maintain a consistent and organised project workflow. It served as the central platform for storing and managing project-related files, including user interview transcripts, feedback documentation, Scrum meeting summaries, and other supporting materials. Its version history and easily accessible functionalities ensured transparency and easy access for all team members and supervisors.

**Microsoft Teams (MS Teams)** [6] - Microsoft Teams was chosen as the primary communication platform due to its convenient and quick meeting scheduling, video conferencing, and file sharing. It was used for Scrum meetings within the team as well as meetings with supervisors. The platform was also used for holding user interviews.

**ChatGPT** [13] - ChatGPT was used as a supportive tool throughout the project. It helped with wording of the thesis, conducting market research, summarising user interviews and feedback, discussing technical approaches, UI design principles, and accessibility considerations.

**Claude** [14] - Claude was primarily used as an AI-assisted development tool, particularly for frontend implementation. It supported resolving code-related issues, exploring optimal technical approaches and making the development process more time-efficient.

## 2.4 Process

The development of Buka applied the following approaches, with the goal of supporting iterative and user-centred software development: the Scrum framework, UX/UI design principles, and design thinking practices.

Scrum enables structured teamwork and continuous feedback through sprint cycles, while UX/UI design focuses on the product's usability and visuals. Design thinking further supports the process by following the steps of problem-solving, prototyping, and continuous improvement.

### 2.4.1 Implementation of Scrum methodology

Scrum is an agile team collaboration framework widely used in software development and big corporations [15]. It enables both small and large teams to structure their work into manageable, smaller units, defined as sprints, which are development cycles with clearly defined goals and a clear deadline. This approach supports continuous progress, adaptability, and receiving regular feedback and validation of outcomes.

A key component of Scrum is the use of recurring meetings, particularly stand-ups, which are short, structured sessions held once or multiple times per week, depending on the project. These meetings allow team members to share progress, discuss challenges, and coordinate ongoing tasks. At the end of each sprint, teams typically hold longer review sessions to present completed work and gather feedback from Scrum Masters.

During the thesis, Scrum principles were applied through a two-week sprint cycle. In our context, supervisors acted as Scrum Masters, and sprint review meetings were held biweekly to review progress, provide guidance and get feedback. Throughout each sprint, the whole team (four people) conducted one to three stand-up meetings per week, during which members presented their work, exchanged ideas, and supported each other in any challenges we faced.

Given that the start-up Buka operated through two parallel teams, the team of the current thesis and development team for the second thesis, collaboration between the teams was adapted accordingly. Each team organised regular co-working sessions between themselves. The team of this thesis primarily held these sessions during weekends, with

scheduling adjusted weekly, while the development team conducted theirs on Wednesdays and Thursdays. Cross-team participation was also encouraged, and members occasionally joined each other's sessions to strengthen collaboration.

Overall, the integration Scrum practices and flexible co-working sessions contributed to improved team coordination, knowledge sharing, and efficient task execution throughout the project.

### 2.4.2 Fundamentals of user experience and user interface design

User experience and user interface design are closely related concepts that are often used together, but they refer to different aspects of product design [16].

User experience describes the overall experience a person has when interacting with a product, system, or service. It includes all aspects of the interaction, such as usability, accessibility, efficiency, and the user's feelings about the product. UX is not only limited to digital products like websites or applications, but it also applies to physical products and services. The main goal of UX design is to focus on the user and ensure that the product is useful and has a positive effect on the user. This involves really understanding the user's needs and analysing how well the product meets those needs [17].

User interface focuses on the visual and interactive elements that users interact with. It is the meeting point of the system and the user. UI design includes the layout, structure, and appearance of all elements of a system, such as buttons, menus, and text. The goal of UI design is to create interfaces that are clear, visually appealing, and easy to navigate, allowing users to complete tasks efficiently [18].

Although UX and UI are different, they work closely together [16]. UX design defines how the product should work and what users need, while UI design brings these ideas to life through visual and interactive design.

When combined effectively, UX and UI design improve user satisfaction and usability. For companies, this can lead to lower costs, higher user retention, and increased revenue, as satisfied users are more likely to continue using the product.

### 2.4.3 Implementation of design thinking practices

Design thinking is a creative, iterative approach to problem-solving that places the user at the center of every decision. It focuses on understanding user needs, pain points, and expectations, and uses those insights to guide the development of products that are useful, feasible, and sustainable [19]. The UX design process builds on this foundation as a structured series of steps that designers follow to transform an idea into a user-friendly product [20]. This process typically includes the following stages: defining the problem, research, analysis and planning, design, prototyping, testing, implementation (launch), and iteration.

The process is important in the development of any product, because it helps designers understand user needs on a deeper level and ensures that the final product is intuitive, effective, and aligned with real-world problems. By following a structured approach, designers can address all key aspects of usability and user satisfaction, helping to ensure that the product will be successful and well thought through.

The first stage, defining the problem, focuses on identifying what needs to be created and why. Designers must clearly understand the problem they are solving and the value the product will provide to users.

The second stage, research, involves gathering information about users, the market, and the context in which the product will be used. This often includes user interviews, surveys, and market analysis, such as conducting market research and competitor analysis.

In the third stage, analysis and planning, designers organise and work with the collected data. They define the overall approach and may create user personas, user stories, and initial wireframes to guide the design process.

The fourth stage, design, focuses on developing initial solutions. Designers begin shaping the user experience and user interface by defining the structure, navigation, layout, usability, and accessibility of the product.

During the prototyping stage, design concepts are turned into interactive models. Prototypes allow designers to simulate how the product will function and prepare it for user testing [19].

In the testing phase, prototypes are tested with real users. Usability testing helps identify problems, gather feedback, and understand the user's perspective, allowing designers to fix the product according to the user's point of view [21].

Once testing is complete and improvements have been made, the product moves to the development stage, where the main goal is to get the product live and make it accessible to users.

The final stage, iteration, is for continuous improvement. Based on user feedback and changing needs, the product is regularly updated and refined to remain relevant and effective [20].

In the context of the startup project Buka, all these stages were followed to ensure that the final product provides a high-quality user experience and meets the needs of its target users.

# 3 Requirements

This chapter outlines the requirements for the development of the Buka system. It begins with a comparison of existing solutions, including Begin [22], Deputy [23], Wemply [24] and RotaCloud [25], as well as traditional spreadsheet, and paper-based approaches. The findings from this analysis, together with market research and user feedback, form the basis for defining Buka's requirements.

The requirements are categorized into business, functional, supporting, and non-functional requirements, providing an overview of both system functionality, business needs and quality attributes.

## 3.1 Alternative solutions and comparison

Before the development of the Buka scheduling system, an analysis of existing workforce management and scheduling solutions was conducted to better understand how current systems address everyday operational challenges and what differentiates them from one another. The analysis focuses on widely used platforms such as Begin, Wemply, Deputy and RotaCloud, as well as traditional spreadsheet- and paper-based solutions. These solutions were selected based on their prevalence in real-world use, as the authors have had direct experience working with them in practical environments.

- **Begin** - an Estonian workforce management platform that is well adapted to local regulations and business practices. Its main strengths include strong localization, integration with payroll and other business systems, and comprehensive functionality, making it suitable for larger organisations. As shown in Figure 1, the platform provides a calendar-based interface, commonly used in traditional workforce management systems.

  However, the system may be too complex for smaller teams and lacks flexibility in dynamic work environments. Additionally, its user interface is often perceived as outdated, which can negatively affect usability and overall user experience.

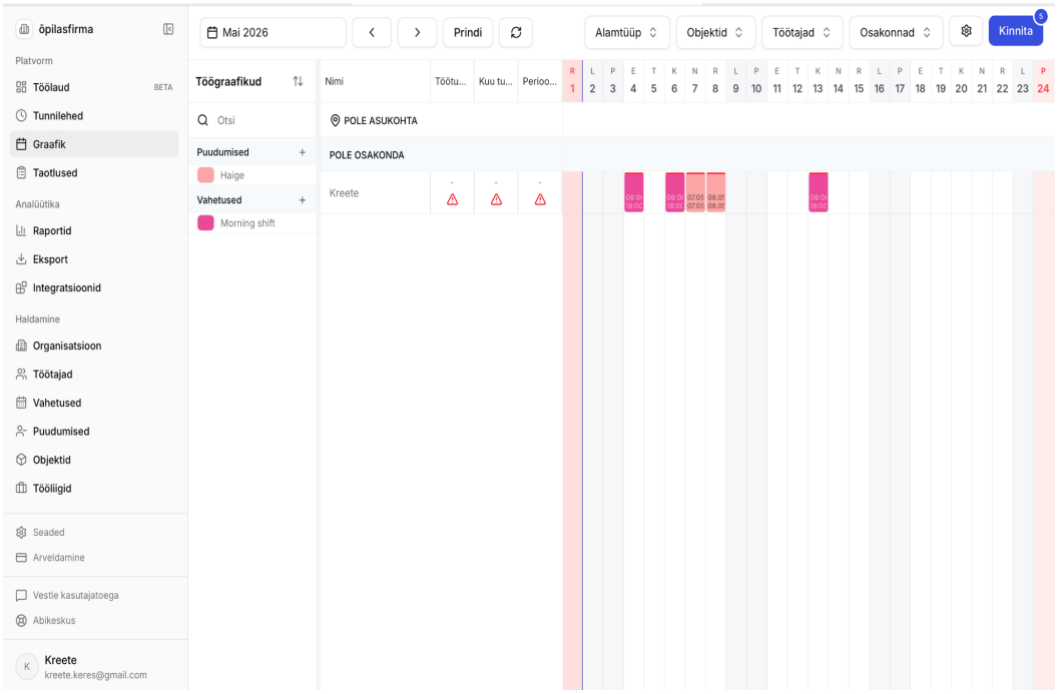


Figure 1. Calendar view of Begin.

- **Deputy** - a widely used Australian workforce management platform designed for employee scheduling, time tracking, and team communication. Its main strengths include an intuitive user interface, strong automation features such as auto scheduling, and mobile accessibility, allowing users to manage shifts in real time. As shown in Figure 2, the platform uses a modern calendar-based scheduling interface with clear shift visualization and team management functionality. The platform is also used in Estonia by hospitality businesses, including establishments within the Siigur Restaurants group, such as POKO.

  However, Deputy also has some limitations. The platform depends on integrations for payroll functionality, which can increase complexity and costs. Additionally, pricing may become relatively high for larger teams, and the system may not fully meet the needs of organisations looking for a complete all-in-one solution.

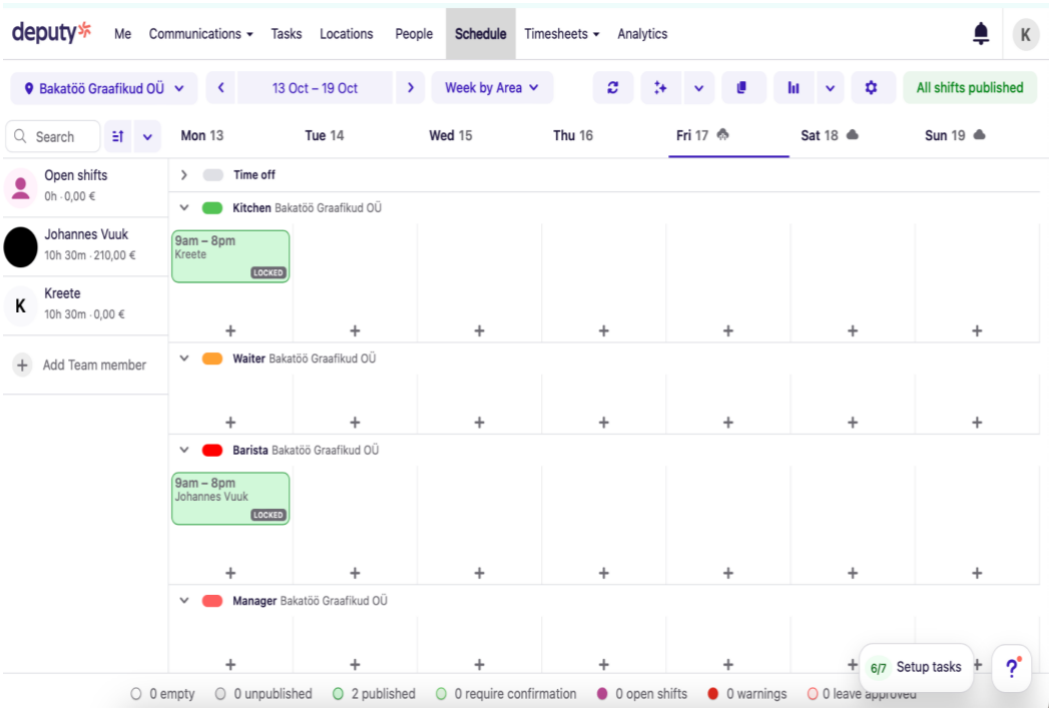


Figure 2. Calendar view of Deputy.

- **Wemply** - an Estonian workforce management platform that provides an all-in-one solution for scheduling, time tracking, and HR management. It is widely recognized for its strong integration with Estonian payroll systems, making it particularly valued by accounting professionals. As shown in Figure 3, Wemply uses a traditional calendar-based scheduling interface, that combines shift planning, events, and absence management within a single view. The platform is suitable for both small and medium-sized businesses and is used by several well-known Estonian companies, including Hesburger, MyFitness, Alexela, and ABB.

  However, Wemply also presents several usability challenges. The user interface is often perceived as outdated and less intuitive, which can make navigation confusing for users. In addition, the platform is not well optimized for mobile devices, resulting in a less convenient user experience for employees who need to manage their schedules on the go.

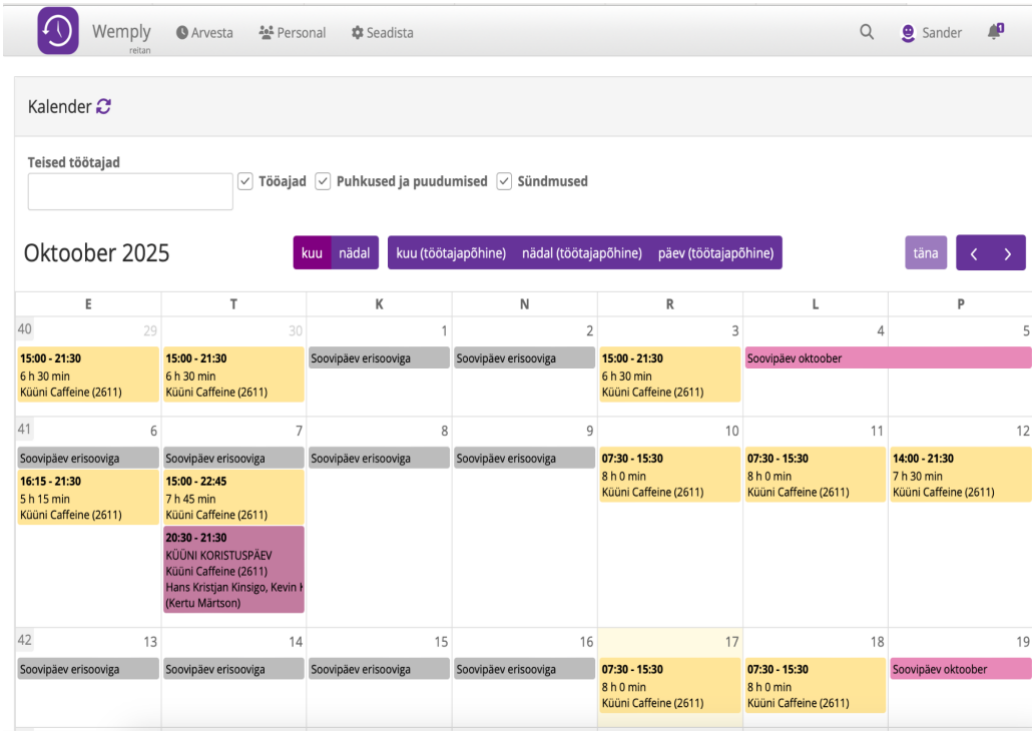


Figure 3. Calendar view of Wemply.

- **RotaCloud** - a UK-based workforce management platform designed for employee scheduling, time tracking, and leave management. Its main strengths include a simple and intuitive interface, drag-and-drop shift planning, and strong mobile accessibility, allowing employees to view schedules, request leave, and manage shifts in real time. As shown in Figure 4, the platform combines schedule management with a mobile-friendly interface, making workforce planning and employee communication more accessible across devices. The platform is particularly suitable for small and medium-sized businesses. In Estonia, the platform is used by some smaller and hospitality-focused businesses, including Paper Mill.

  However, while it is easy to use, it may lack more advanced HR and payroll integration features required by larger organisations. Additionally, customization options can be limited, and some functionalities are better suited for standard scheduling needs rather than complex operational environments.

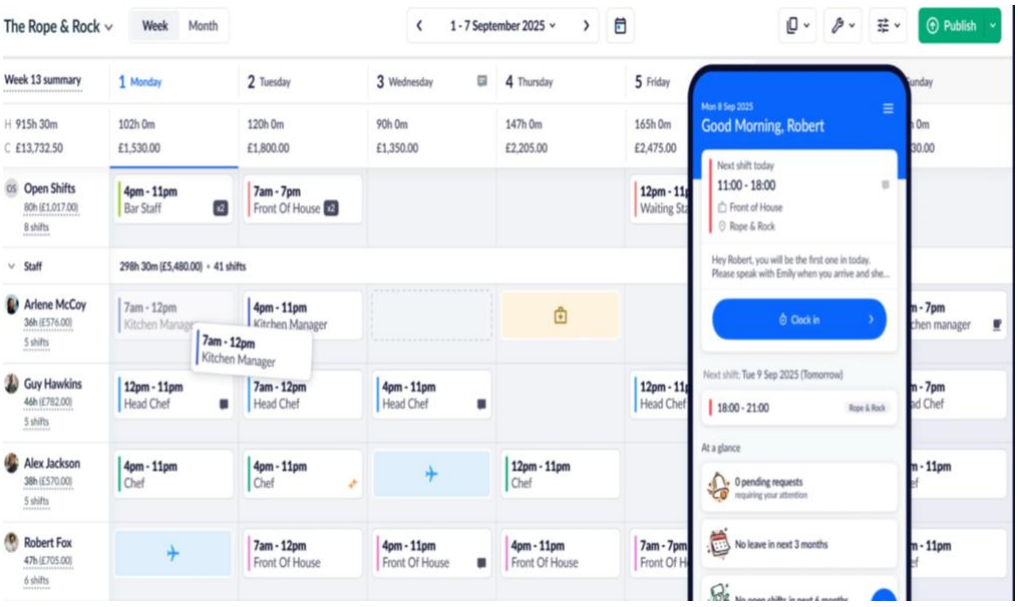


Figure 4. Calendar view of RotaCloud.

- **Paper-based and Excel-based solutions** - traditional paper-based and spreadsheet-based solutions, such as Microsoft Excel, are widely used for workforce scheduling due to their accessibility, low cost, and ease of initial setup. They allow a high degree of flexibility and can be easily adapted to different organisational needs without requiring specialized software.

  However, these approaches also have significant limitations. They are highly prone to human error, lack automation, and do not support real-time updates or efficient communication between employees and managers. Additionally, managing schedules becomes increasingly time-consuming and difficult as the organisation grows, making these solutions poorly scalable and less suitable for dynamic and fast-paced work environments.

The comparison focused on key aspects such as usability, level of automation, flexibility in handling changes, and overall user experience. Special attention was given to how well each solution supports the needs of small and medium-sized enterprises. The results of this comparison are presented in Table 1.

### 3.1.1 Comparative analysis

Table 1 presents a comparison of key features in popular workforce management systems used in Estonian hospitality companies.

Table 1. Comparative analysis.

| Feature | Begin | Wemply | Deputy | RotaCloud | Excel/Paper |
|---|---|---|---|---|---|
| Pricing | Approx. 5-10€ per month per user | 7€ per user per month (standard) | $6.5 per user per month (core) | £10 per month (up to 5 employees) | $6 per user per month (business basic) |
| Automatic schedule generation | No | No | Yes | No | No |
| Automated schedule adjustments | No | No | Partial | No | No |
| Automated and fair workload balancing | No | No | Yes | No | No |
| Manual editing | Yes | Yes | Yes | Yes | Yes |
| Employee availability input (vacations, sick leaves, days off) | Yes | Yes | Yes | Yes | Partial (verbally) |
| Work hours tracking in real time | Yes | Yes | Yes | Yes | Partial |
| Salary tracking in real time | No | No | Yes (via integrations) | Partial | No |
| Document management and storage | Yes | Yes | Partial | No | No |
| Automatic shift-swapping between employees | No | No | No | No | No |

The table shows that none of the analysed systems fully support all key automation and workforce management features. Deputy stands out by offering automatic schedule generation, partial automated adjustments, and fair workload balancing, making it the most advanced in terms of automation. However, it relies on integrations for salary tracking and does not provide a fully unified solution. Begin and Wemply provide strong support for HR-related functionalities, such as document management and work hour tracking, but lack automation and dynamic scheduling capabilities. RotaCloud offers a simple and user-friendly interface, but its functionality is limited when handling more complex scheduling scenarios.

In contrast, Excel, and paper-based solutions offer flexibility and ease of use but lack automation, real-time capabilities, and scalability.

Overall, the comparison highlights a clear gap between complex, automation-focused systems and simpler, manual solutions. This indicates the need for a system that effectively combines ease of use with essential automation features, particularly in handling dynamic and last-minute schedule changes, ensuring fair workload distribution, and providing a transparent and easy-to-understand wage calculation system.

## 3.2 Key insights based on competitor analysis and user feedback

The requirements for the Buka system are derived from the previously conducted competitor analysis and user feedback. The analysis revealed several limitations in existing workforce management solutions, including a lack of automation, poor usability, limited flexibility in handling dynamic scheduling changes, and insufficient transparency in wage calculation.

In addition to competitor analysis, user feedback was collected to better understand real-world challenges faced by employees and company managers in scheduling processes. These insights helped identify the most critical pain points and user needs, which form the foundation for defining the system requirements.

Main key insights:

- Many existing systems are too complex for small and medium-sized businesses.
- Users value simplicity and intuitive user interfaces.
- There is a strong need for automation in schedule creation and adjustments.
- Handling last-minute changes (e.g. sick leaves) is currently inefficient.
- Employees want more control and transparency over their schedules.
- There is a lack of clear and understandable wage calculation systems integrated into schedule management systems.
- Mobile usability is often poor or limited in existing solutions.
- Manual solutions (Excel, paper) are still widely used, but very inefficient and time consuming.

- Creating new schedules is often time-consuming and stressful for company managers.
- Many functionalities still require significant manual effort, even though they could be automated to improve efficiency and reduce workload.
- Workers frequently rely on multiple communication platforms to manage shift changes, coverage, and related coordination, even though these processes could be more efficiently handled within a single, centralised system.

Based on these findings, the Buka system aims to address the key shortcomings of existing solutions by providing a simple, user-friendly, and centralised workforce management platform. It focuses on reducing manual work through automation, improving handling of dynamic schedule changes, ensuring fair workload distribution, and increasing transparency in work hours and wage calculation. Additionally, the system aims to improve mobile usability and centralise communication related to shift management.

### 3.2.1 Business requirements

Business requirements define the strategic direction of a project by describing the key business needs and objectives that a solution must address [26]. They focus on the "what" and "why" of the project rather than the technical implementation, ensuring alignment with organisational goals. According to Jama Software [26] and AltexSoft [27], business requirements outline what the organisation expects from the system, highlight its value proposition, and establish measurable criteria for evaluating success. Their main purpose is to provide a clear, high-level overview of the project's goals and expected business outcomes.

The business requirements for the development of the software application Buka are as follows:

- Operational efficiency - the company needs to reduce time and effort spent on managing employee schedules, working hours, and leave in order to improve overall operational efficiency.
- Cost optimization - the company needs to better control labour costs by improving visibility into working hours, overtime, and staffing levels.

- Employee autonomy - the company needs to empower employees to manage their own work-related requests independently to reduce reliance on managers and improve workflow.
- Employee satisfaction - the company needs to improve the overall employee experience related to scheduling and leave management to increase satisfaction.
- Organisational transparency - the company needs clear and consistent visibility into schedules, workloads, and employee availability to ensure fairness and informed decision-making.
- Effective workforce planning - the company needs to improve workforce planning capabilities to ensure the right number of employees are scheduled at the right time.
- Manual work reduction - the company needs to minimise manual administrative tasks related to scheduling, coordination, and communication.
- Data consistency and accessibility - the company needs a reliable and consistent way to manage and access employee-related data across the organisation.
- Error reduction - the company needs to reduce human errors in tracking working hours and managing payroll-related information.
- Operational continuity - the company needs to ensure that all required shifts are covered to maintain uninterrupted business operations.
- Regulatory compliance - the company needs to ensure that employee data and work-hour tracking comply with applicable legal and data protection regulations (e.g., GDPR).

### 3.2.2 Functional requirements

Functional requirements define a system's main features and functions [28], [29]. In other words, they describe what exactly the software product must do under normal conditions to meet the user's needs. Functional requirements describe both product features and nice-to-haves that can enhance user experience but aren't necessary for the system to function.

The functional requirements for the development of the software application Buka are as follows:

- User registration - the system shall allow new users to create an account.

- User authentication - the system shall allow registered users to log in to their accounts.
- Profile management - the system shall allow users to manage their personal profiles and account information.
- Role-based access control - the system shall provide different access levels (e.g., employee, manager, company owner) to ensure each user has permissions appropriate to their role.
- Employee data management - the system shall store and manage employee-related data, including roles, contract details, and workload information.
- Employee document management - the system shall allow for the upload, storage, and management of relevant employee documents (e.g., health certificates, student certificates).
- Schedule viewing - the system shall allow users to view their personal work schedules in daily, weekly, and monthly formats.
- Team calendar view - the system shall provide a team-based calendar view displaying shifts and other relevant information.
- External calendar integration - the system shall allow users to integrate their work schedules with external calendar applications (e.g., Google Calendar).
- Open shifts management - the system shall enable users to release their assigned or sick leave shifts to an “Open Shifts” section, making them visible and available for other users to claim.
- SOS shift management - the system shall notify users 48 hours before the start of an SOS shift, allowing them sufficient time to volunteer to cover the shift.
- Shift swap requests - the system shall allow users to initiate and submit requests to swap shifts with coworkers.
- Automated shift matching - the system shall automatically identify and suggest suitable replacements for a shift based on employee availability, workload, and qualifications.
- Leave request submission - the system shall allow users to submit requests for vacation, sick leave, and other days off.
- Leave request tracking - the system shall allow users to view the status (pending, approved, rejected) of their leave requests.
- Leave request withdrawal - the system shall allow users to withdraw their previously submitted leave requests.

- Working hours tracking - the system shall automatically track and log real-time working hours for each user.
- Overtime/undertime calculation - the system shall calculate overtime and undertime based on tracked working hours.
- Payroll estimation - the system shall automatically calculate and display an estimated salary based on working hours, contract details, and applicable regulations.
- Real-time notifications - the system shall send users real-time notifications for important events, including schedule changes, relevant requests, and approaching deadlines.

### 3.2.3 Supporting requirements

Supporting requirements, similar to functional requirements, define the core system functionalities [30]. However, supporting requirements focus more on quality attributes such as usability, performance, reliability, and overall user experience. In contrast to core functional requirements, it emphasizes more on aspects like flexibility, ease of use, and system integration.

These requirements support and extend the core system functionality by defining additional features that improve usability and efficiency, without being a part of forming the main system logic.

In the context of this thesis, supporting requirements are defined for both the company managers and worker views of the Buka system.

**Supporting requirements for the company manager view**

- Employee onboarding and management - the system shall allow company managers to invite new employees, review and approve join requests, and manage employee profiles, including contract-based information that can be updated when necessary.
- Employee data and document oversight - the system shall provide company managers with a comprehensive overview of all employee-related data and documents, including contracts, certifications, and other relevant records.

- Compensation and bonus management - the system shall enable company managers to assign and manage bonuses for employees and themselves, based on hourly or monthly bonus schemes.
- Organisational structuring and filtering - the system shall allow company managers to organise and filter data based on company structure (e.g., different cafés or locations), including employees, schedules, and related information.
- Notification and awareness system - the system shall notify company managers of key activities and events, including join requests, shift swaps, leave requests, days off requests, shift claim and shift drop requests, profile changes, document uploads, and upcoming deadlines.
- Approval and decision-making support - the system shall enable company managers to review and approve or reject various employee actions, including join requests, shift swaps, leave requests, and shift claims or drops.
- Scheduling oversight and control - the system shall provide company managers with a draft view of generated schedules, enabling visibility into scheduling gaps (e.g., uncovered shifts) before publication, as well as the ability to create, modify, and delete shifts as needed.
- Emergency shift management - the system shall allow company managers to mark shifts as urgent (e.g., SOS shifts) and notify available employees to ensure quick coverage.
- Workforce insights and tracking - the system shall provide company managers with insights into employee activity, including submitted availability (preferred days), working hours, and overtime/undertime across monthly, quarterly, and yearly views.
- System configuration and customization - the system shall allow company managers to configure system settings, including scheduling rules, deadlines, workforce limits, salary settings, visual customization (e.g., color-coding locations), and company-related information.
- Audit trail and activity history - the system shall maintain a history of administrative actions to ensure transparency and allow tracking of all changes made within the system.

**Supporting requirements for the worker view**

- Shift management flexibility - the system shall allow employees to submit shift swap requests, publish shifts in the “up for grabs” section, respond to available shifts (including SOS shifts), and withdraw previously submitted requests or postings.
- Salary transparency - the system shall allow employees to access their salary information, including detailed breakdowns and any personalized bonus-related comments provided by the company manager.
- Notifications and reminders - the system shall notify employees of relevant events and approaching deadlines, including shift swaps, rest day submissions, and newly available or urgent (SOS) shifts.
- Request tracking and visibility - the system shall allow employees to view the status of all submitted requests (e.g., pending, approved, or rejected) related to shifts, vacations, and other absences.

### 3.2.4 Non-functional requirements

Non-functional requirements, are a set of specifications that describe the system’s operation capabilities and constraints [31], [32]. Their main goal is to specify criteria that can be used to judge the operation of a system, rather than specific behaviours. These are basically requirements that focus more on the user expectations and user experience, outlining how well a system operates, and describing how the product works. Some examples of non-functional requirements include performance, usability, security, scalability and availability.

In the context of Buka, these requirements are more applicable to the companion thesis [33].

The non-functional requirements for the Buka system are as follows:

- Performance - the system shall respond to user actions (e.g., loading schedules, submitting requests) within $\leq 2$ seconds under normal operating conditions.
- Availability - the system shall be available at least 99% of the time, excluding scheduled maintenance.

- Usability - the system shall provide an intuitive and user-friendly interface, allowing new users to complete basic tasks (e.g., viewing schedule, submitting requests) without prior training.
- Scalability - the system shall support an increasing number of users, employees, and locations (e.g., cafés) without significant performance degradation.
- Security - the system shall ensure secure authentication, role-based access control, and protection of sensitive user data.
- Data Protection (GDPR Compliance) - the system shall comply with GDPR by ensuring that personal data is securely stored, processed only when necessary, and accessible based on user roles.
- Reliability - the system shall ensure consistent and error-free operation, minimising system crashes and preventing data loss.
- Maintainability - the system shall be designed in a way that allows easy updates, bug fixes, and future feature extensions.
- Compatibility - the system shall be accessible across different devices (desktop, tablet, mobile) and major web browsers.
- Response to real-time updates - the system shall reflect schedule changes, requests, and updates in real time or with minimal delay ($\leq 2$ seconds).

# 4 UX/UI design and development process

The following chapter provides an overview of the UX/UI design and development process used for the creation of Buka. It describes the different stages of the design workflow, including user research, creation of user personas, prototyping, and visual design decisions. Both low-fidelity and high-fidelity prototypes are presented to demonstrate the evolution of Buka throughout the process.

Emphasis is placed on usability testing and continuous user feedback, as these played a critical role in validating design decisions and improving the final solution according to real user needs and expectations.

## 4.1 User personas

User personas are fictional representations of target users created based on user research, goals, and behaviours. In UX design, personas help designers better understand user needs and support more user-centred design decisions [34].

In the context of Buka, personas were created to better understand the needs and challenges of service industry workers and managers.

**Employee user persona (Lisette)**


Figure 5. Employee user persona.

Background:

Lisette, as illustrated in Figure 5, is a young person at the end of high school or at the start of university studies. She is energetic and social, enjoys parties and meeting friends, and tries to live life to the fullest while studying. Working at a café is mainly a way for her to earn extra money rather than a long-term career choice.

Lisette's goals:

1. Earning extra money to cover everyday expenses.
2. Saving money for travelling and new experiences.
3. Affording small pleasures in life.
4. Finding a job that is flexible and fits her busy lifestyle.

Lisette's pain points:

1. Feeling that the salary does not meet her expectations.
2. Unexpected or unclear shifts cause her stress and disrupt personal plans.
3. Lacking clarity. For her, it is difficult to know in advance when she has free days.
4. Feeling that work and free time are not well balanced.

Quote:

"Why is everything so unclear and time-consuming? I want to make my plans early, but unclear work management gets in the way."

**Company manager user persona (Karin)**

Karin, as illustrated in Figure 6, represents a manager (e.g., restaurant manager) who is responsible for creating schedules, managing employees, and handling everyday tasks such as shift swaps, vacation requests, and sick leave requests. She works in a fast-paced environment where quick decisions must be made and clear communication is important.


Figure 6. Company manager user persona.

Karin's goals:

1. Creating and managing work schedules.
2. Handling employee requests quickly and accurately.
3. Ensuring that all shifts are covered.
4. Ensuring that the company's goals are achieved.
5. Maintaining a clear and transparent communication with employees.

Karin's pain points:

1. Scheduling takes a lot of time and effort.
2. A lot of last-minute tasks, that need to be handled quickly.
3. Communication is spread across multiple platforms (Messenger, WhatsApp, etc.)
4. High risk of errors in tracking hours, shifts and requests.
5. Lack of a centralised system for managing all information at once.

Quote:

"I spend too much time managing schedules and answering messages. It would be much easier if everything I needed was all together in one system."

## 4.2 Prototyping and iterative design

Prototyping and iterative design were very valuable and core parts of this project. They were used to help show initial ideas, test and improve them based on user feedback, before developing the final solution.

The process was repeated many times with the goal to improve it after every user feedback, usability testing, and client interviews. This way of working helped ensure that the final solution was based on real user needs and tested ideas, not just initial assumptions.

### 4.2.1 Low-fidelity prototype

At the beginning of the design process, low-fidelity prototypes were created to quickly visualize the system structure, user flows, and core functionalities before focusing on visual design details. Lo-fi prototyping helped test early ideas, identify usability issues, and gather feedback efficiently during the early development stages [35].

1. “Shifts” (main calendar view) page.

As illustrated in Figure 7, the “Shifts” page represents the core functionality of the Buka system, providing a calendar-based overview of all scheduled shifts. Users can view shifts across different timeframes (e.g., monthly view) and coworkers and cafes.

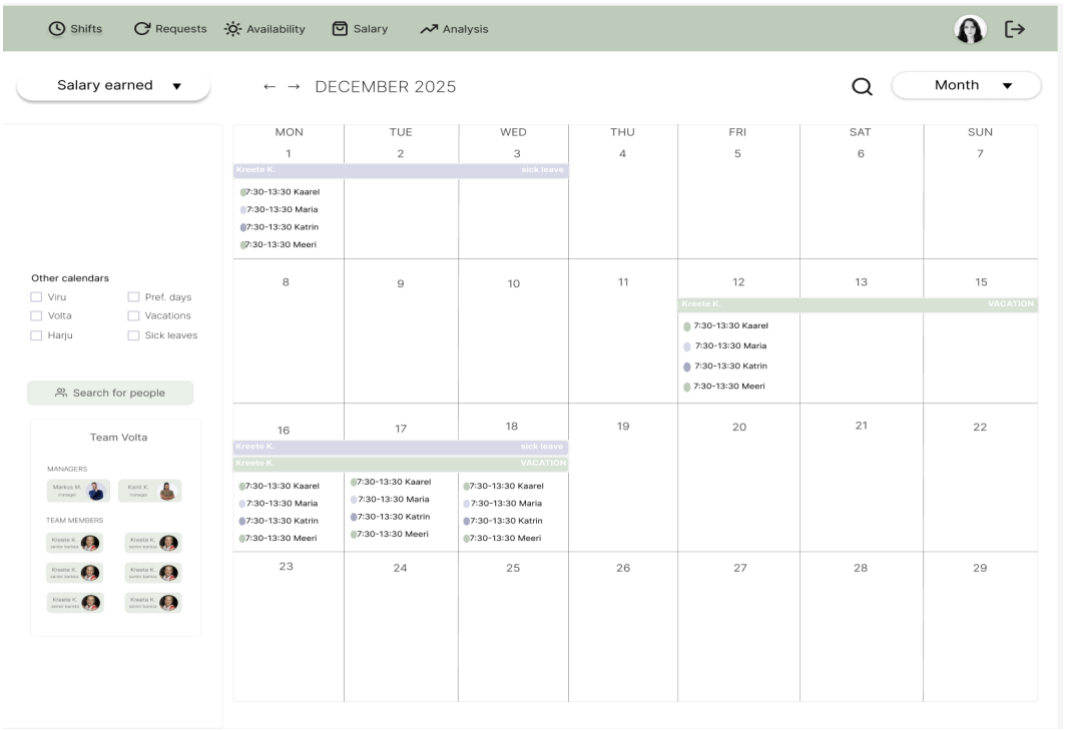


Figure 7. Lo-fi prototype of Buka's calendar view.

2. “Requests” page.

As illustrated in Figure 8, the “Requests” page is designed to manage and track all shift swap requests.

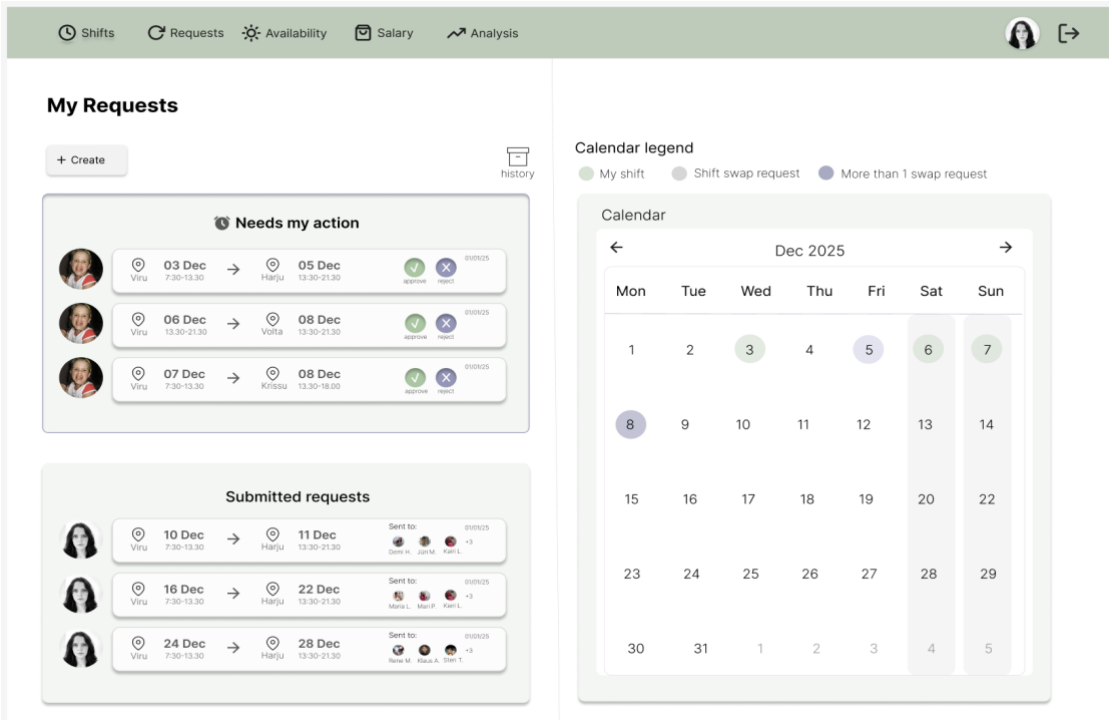


Figure 8. Lo-fi prototype of Buka's requests view.

3. “Availability” page.

As shown in Figure 9, the “Availability” page allows users to submit and manage their availability preferences, including sick leave, vacations, and preferred working days. Requests are visually categorized (e.g., pending and accepted).

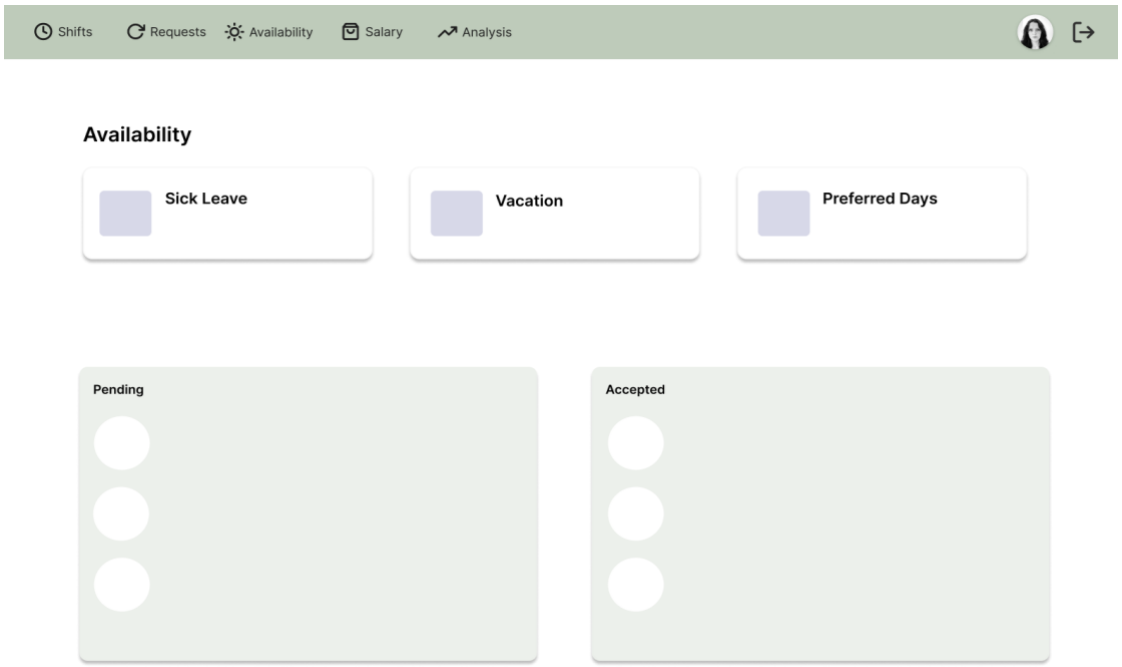


Figure 9. Lo-fi prototype of Buka's availability view.

4. “My profile” page.

As illustrated in Figure 10, the “My Profile” page provides a centralised view of all relevant user information, including personal details, work-related data, contract information, and vacation balance.

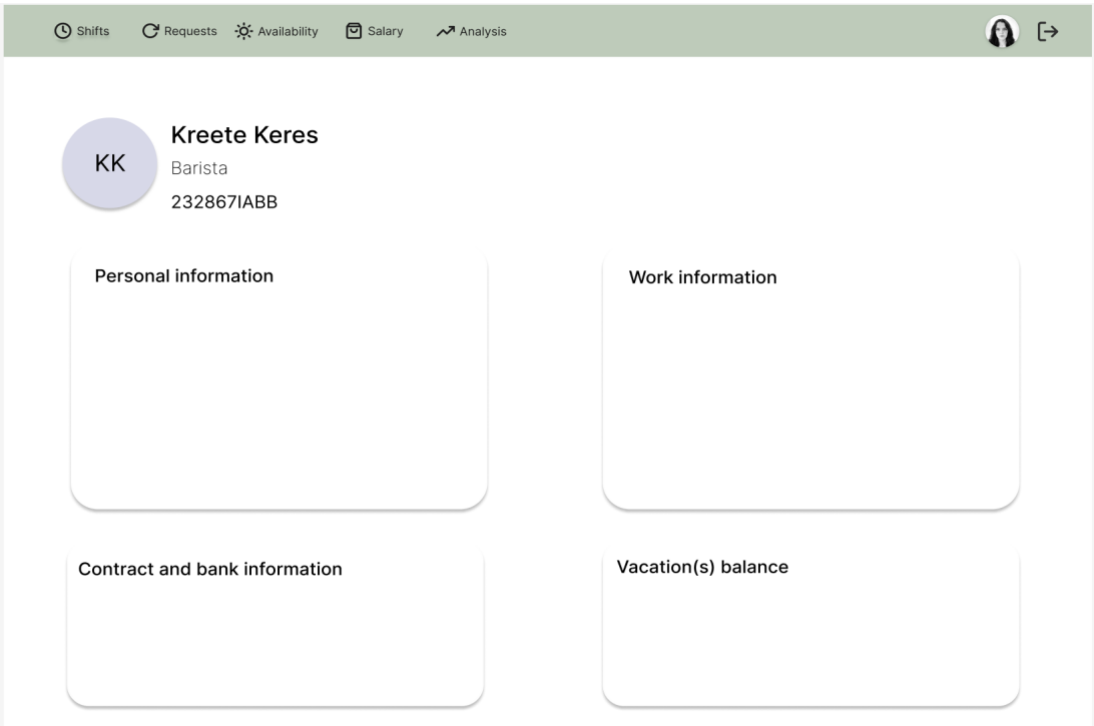


Figure 10. Lo-fi prototype of Buka's user profile view.

5. “Salary” page.

As shown in Figure 11, the “Salary” page offers a detailed overview of the user’s working hours, earnings, and leave usage. Its main goal is to provide users with a transparent salary information overview.

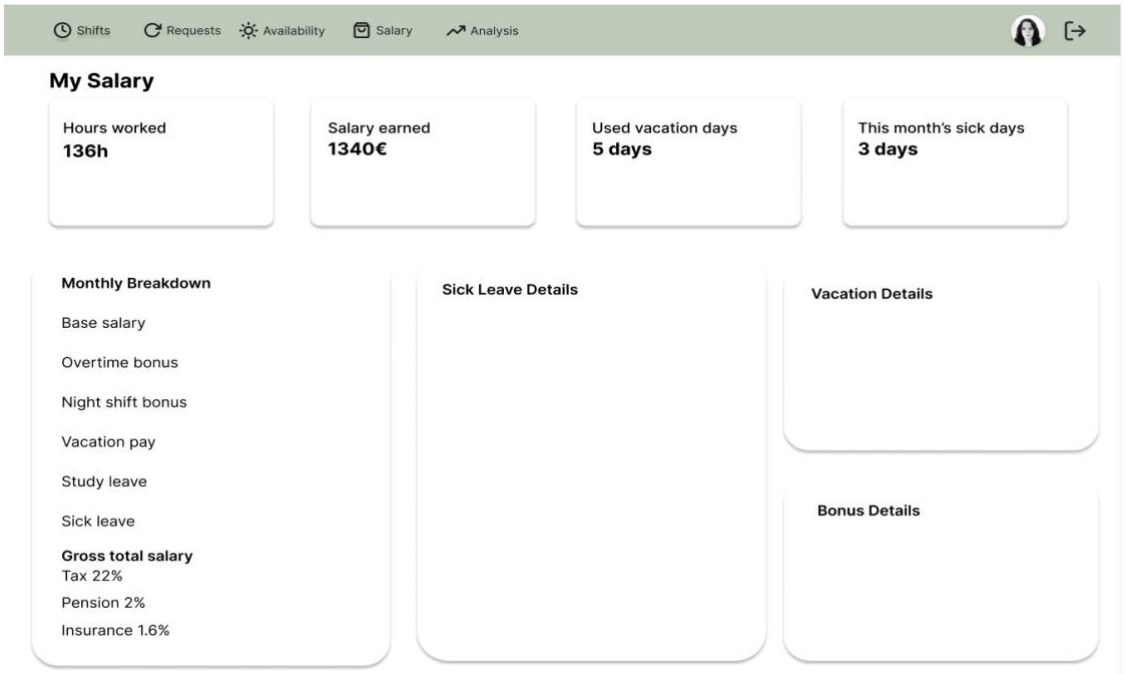


Figure 11. Lo-fi prototype of Buka's salary view.

6. “Login” page.

As illustrated in Figure 12, the “Login” page provides users with a simple authentication interface for accessing Buka’s system.


Figure 12. Lo-fi prototype of Buka's login view.

After that, initial prototypes were tested with users and based on feedback, developed into high-fidelity (hi-fi) prototypes.

### 4.2.2 High-fidelity prototype

High-fidelity prototypes were then developed to create a more realistic representation of the final solution. Compared to lo-fi prototypes, hi-fi prototypes focus more on visual design, interface details, interactions, and overall user experience. In the context of Buka, they were used to further validate the system with real users and improve the final design before implementation [36].

1. “Login” Page.

As illustrated in Figure 13, the hi-fi “Login” page provides users with a modern and visually refined authentication interface for accessing the Buka platform.

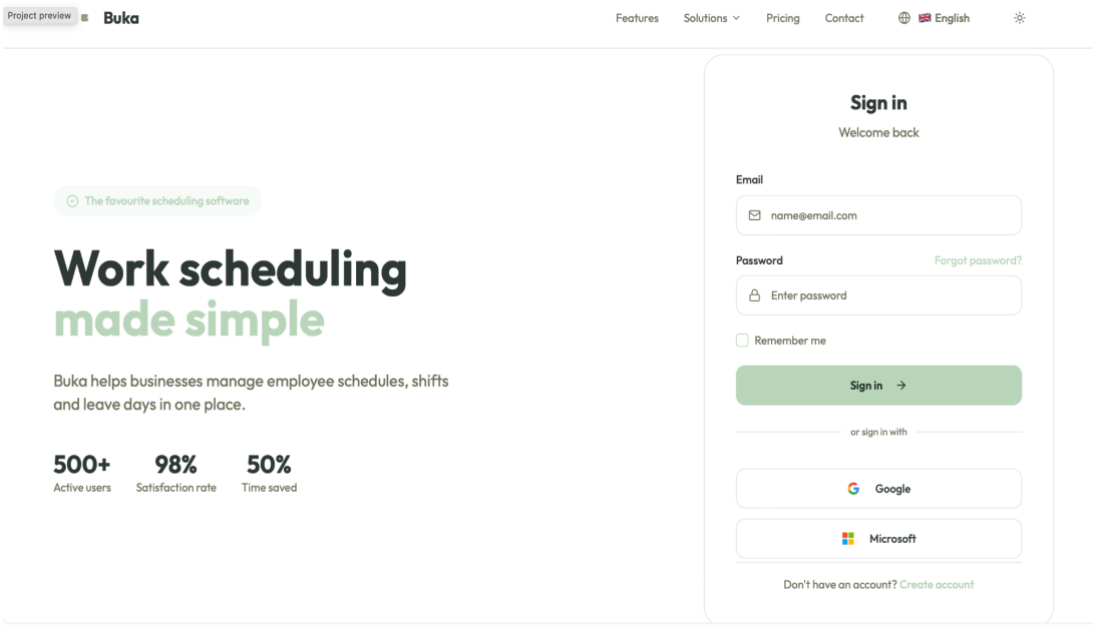


Figure 13. Hi-fi prototype of Buka's login view.

2. “Shifts” page.

As illustrated in Figure 14, the “Shifts” page is the main view of the system, where most scheduling-related actions can be performed. The page provides users with a centralised calendar interface for viewing shifts, coworkers and work locations.

Figure 15 further demonstrates the contextual pop-up interface used for creating requests without leaving the main schedule page. Users can create shift swap, vacation, sick leave, and day-off requests directly from the calendar. This approach reduces the need for separate pages.

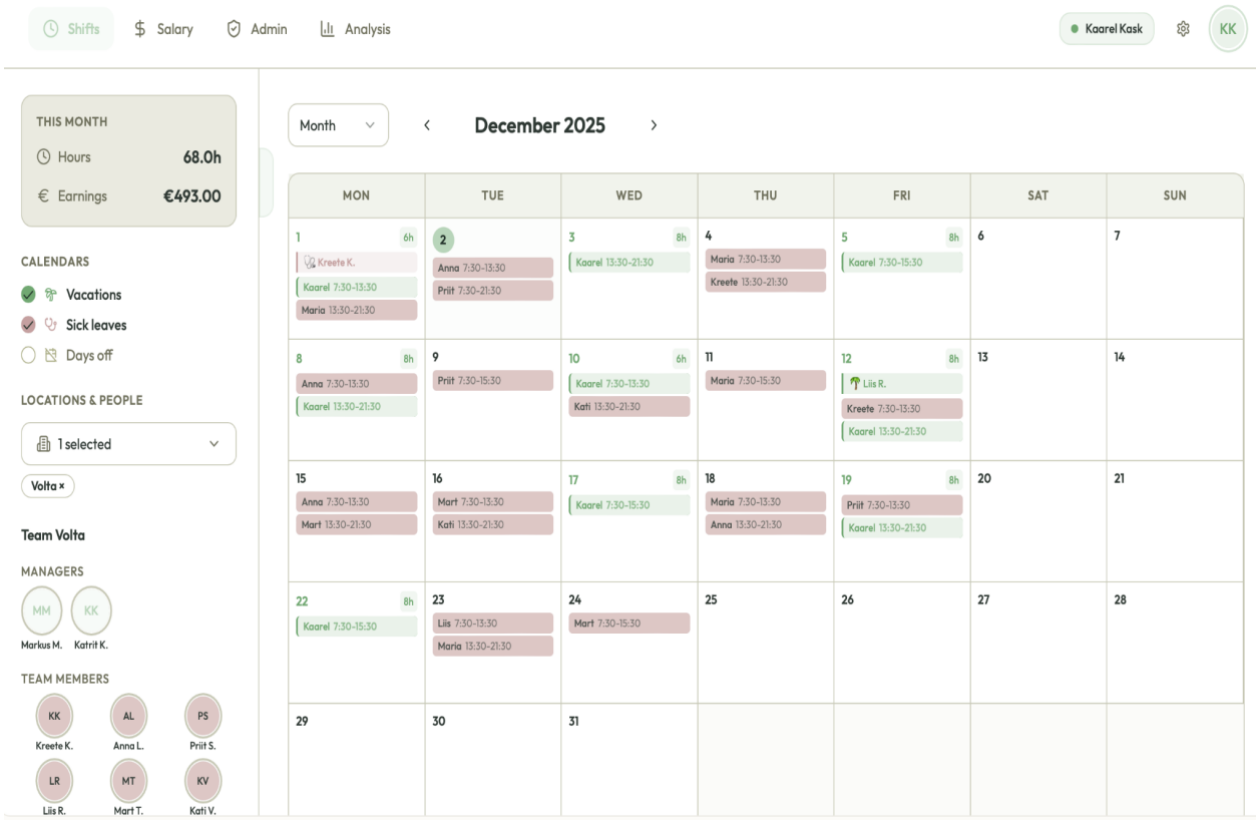


Figure 14. Hi-fi prototype of Buka's calendar view.

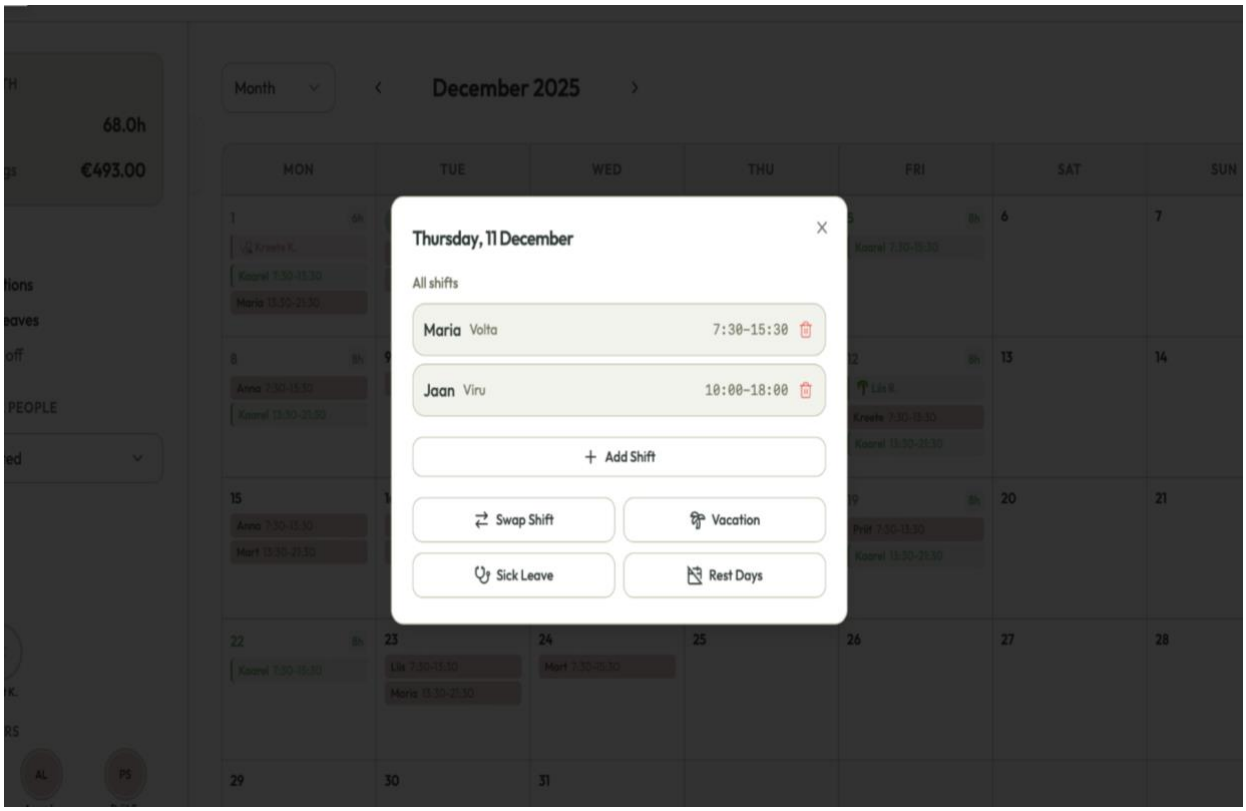


Figure 15. Hi-fi pop-up of shift management actions.

3. “Salary” page

As illustrated in Figure 16, the “Salary” page includes one new feature compared to the lo-fi version – the ability to view and download the work contract as a PDF. The page also provides users with an overview of all salary-related information.

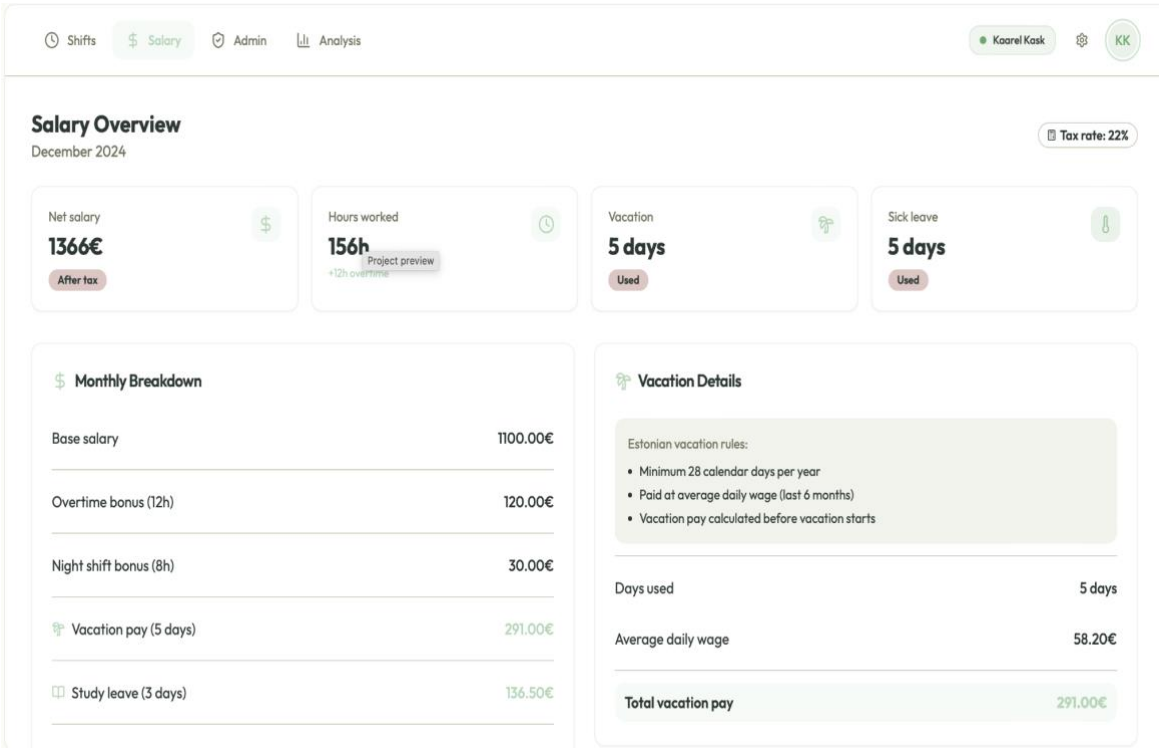


Figure 16. Hi-fi prototype of salary view.

4. “Admin” page

As illustrated in Figures 17 and 18, the “Admin” page brings all admin tasks into one place. Instead of separate pages, managers can now manage join requests, shift swap requests, leave requests, and employee data change requests within the same page.

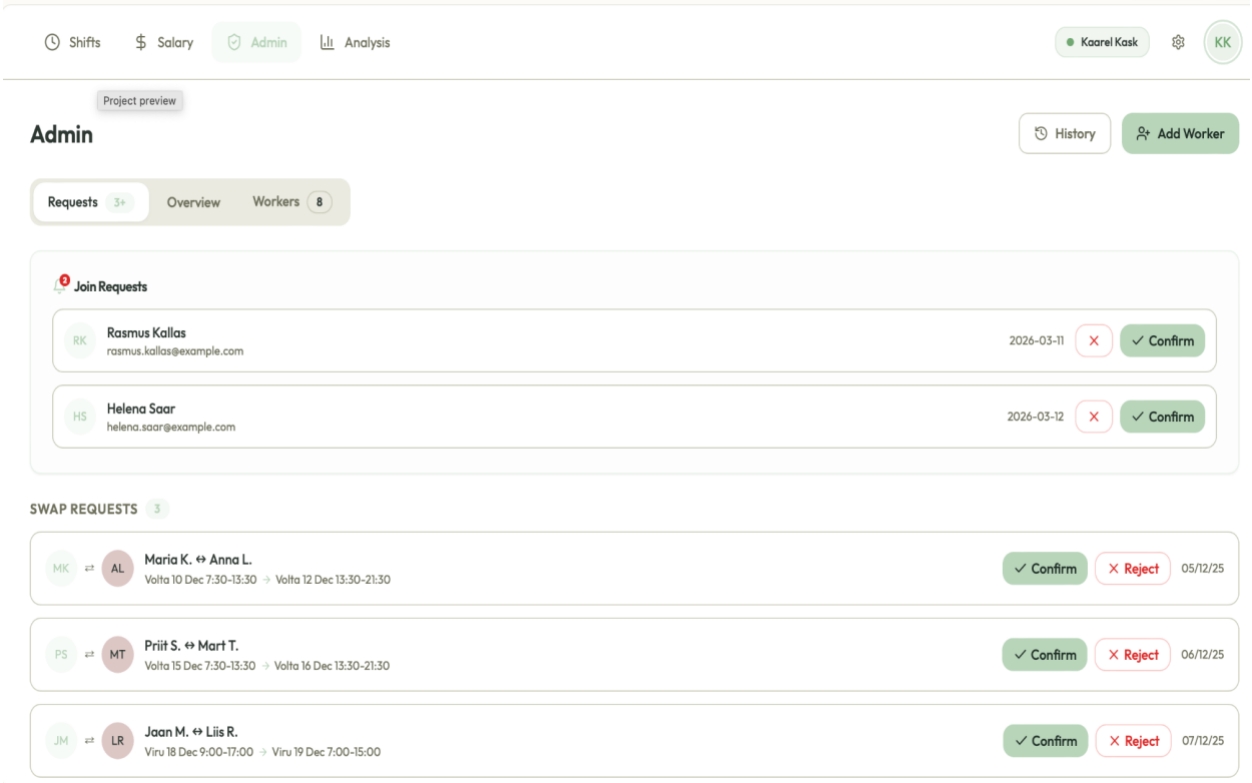


Figure 17. Hi-fi prototype of admin view 1.

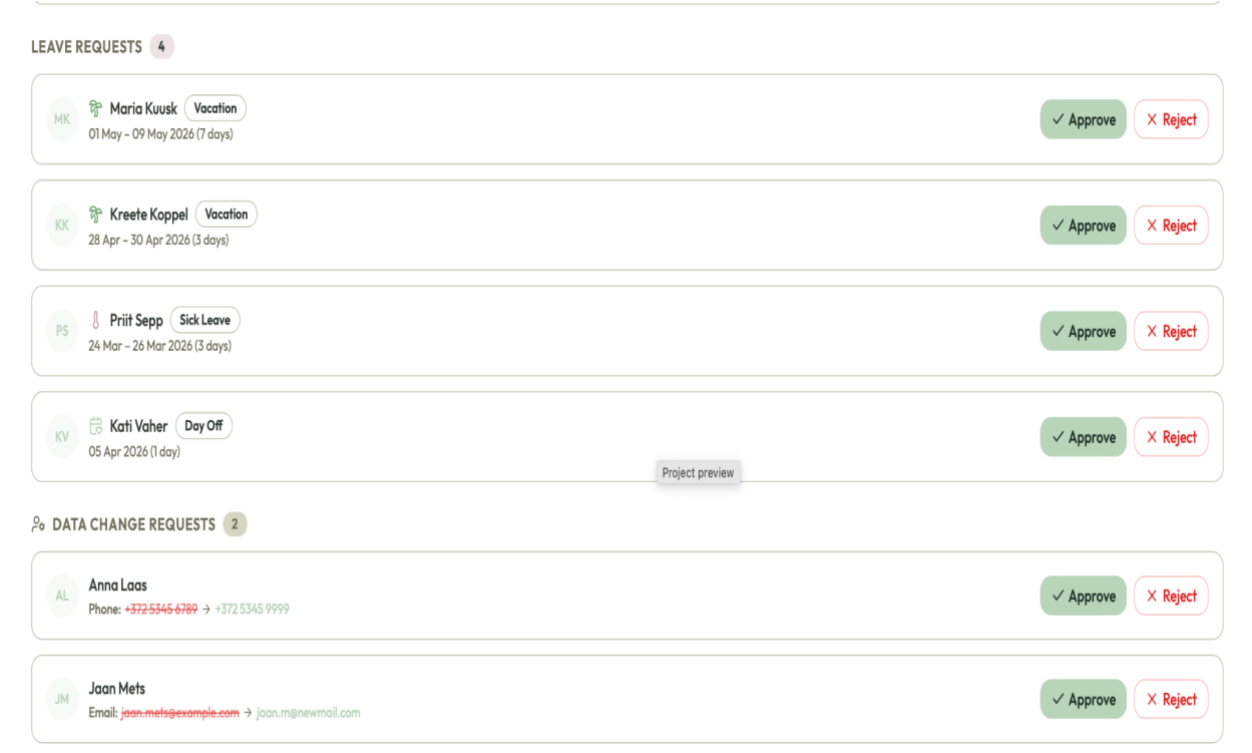


Figure 18. Hi-fi prototype of admin view 2.

5. “My profile” page

As shown in Figure 19, the “My Profile” page now includes options to change the password and enable two-factor authentication. It also shows all personal, work, and document-related information in one place.

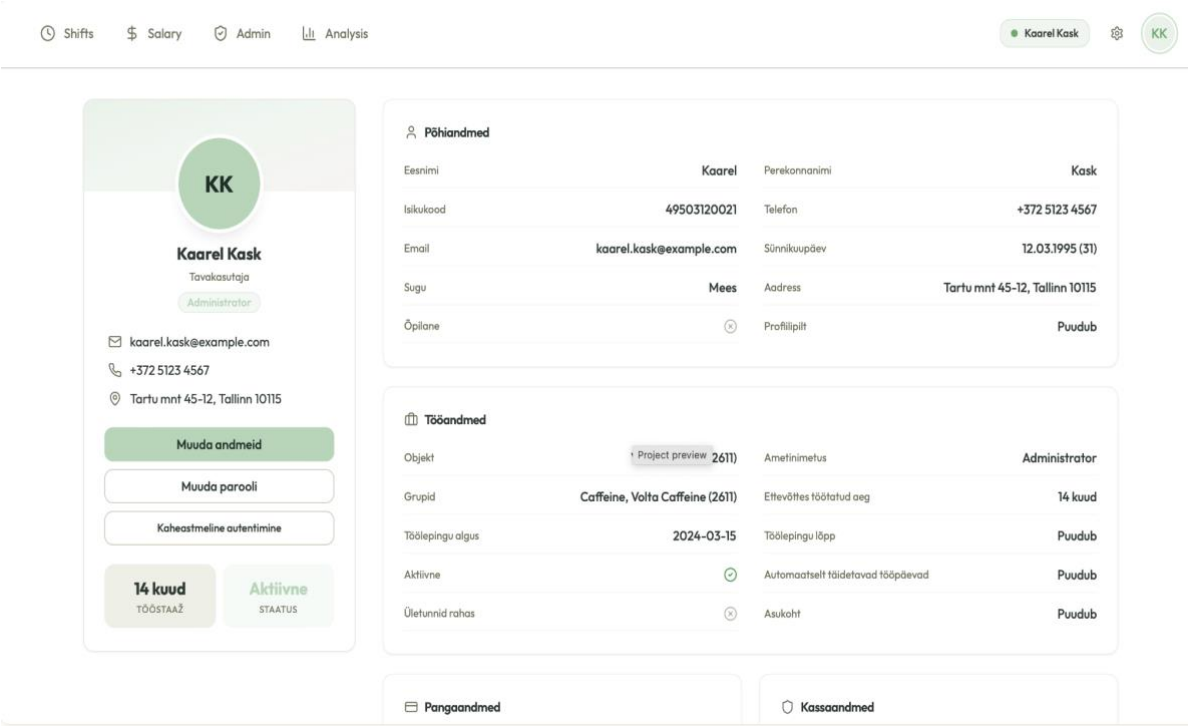


Figure 19. Hi-fi prototype of user's profile view.

## 4.3 Visual design and style decisions

The visual design and style decisions of the system were primarily based on understanding the needs and expectations of end users. This was achieved through UX interviews with potential users and discussions with the lecturer of the User Interface and Application Design course, both in the beginning of the product development process and at the end as well.

Initially, the goal was to create a simple, clean, and minimalistic interface. Light green and light purple were chosen as primary colours to achieve a calm and easy-to-use design. However, feedback showed that these colours lacked contrast and did not highlight important elements clearly. Therefore, the colour palette was updated to a more vibrant

and structured system, based on sage and teal tones, complemented by clearly distinguishable status colours.

The final colour system is built around a five-colour sage and teal palette, ranging from light sage (#CAD2C5) to deep teal (#2F3E46). The medium sage tone (#84A98C) is used as the main accent colour throughout the interface, while darker sage and teal tones are used as secondary colours to create contrast and visual hierarchy.

To ensure clear recognition of different schedule-related events, the system uses fixed status colours: green for vacations, blue for sick leaves, and purple for days off. Each café location is also assigned its own distinct colour from a six-tone palette, which can be customized by managers. Fixed colours help both managers and employees quickly understand the calendar view and provide a clearer overview of schedules, locations, and different shift-related events.

The UI design was mainly inspired by existing calendar and scheduling systems, with a strong focus on clarity and simplicity. Over time, the design evolved based on continuous user feedback and newly added features. As a result, the interface became slightly more complex, but also more functional and useful. The final solution combines a minimalistic and modern design with a clear and practical user experience.

## 4.4 User testing, validation, and iteration

User testing and UX validation played a central role throughout the development of Buka. The purpose of these interviews was to validate design decisions, identify usability issues, and continuously improve the application based on feedback. Testing was conducted iteratively during all stages of development.

The testing process involved both service-sector employees and managers in order to gather feedback from different user perspectives. In addition to end-user testing, the final application was also reviewed by a professional UX designer. The following chapters describe the interview process, key findings, and the improvements implemented based on the collected feedback.

### 4.4.1 Interview process and insights

The first UX interviews were conducted during the autumn semester using the initial idea and lo-fi prototypes. This testing phase involved ten participants, mainly acquaintances and students connected to the TalTech User Interface and Application Design course. The purpose of these interviews was to better understand user expectations, workflows, and pain points related to workforce management systems.

The second round of UX interviews was conducted remotely through Microsoft Teams using a shareable link to the hi-fi prototype created in Lovable. This testing phase involved four participants with previous or current experience in the hospitality and service industry. Two participants were current service-industry employees, while two were former café managers from Caffeine and POKO.

During the interviews, participants were asked to complete predefined tasks within the system, including logging in, managing shifts, and accessing salary information. In addition, participants were asked to identify two functionalities they considered the most valuable and two aspects they believed required improvement. A more detailed overview of the interview questions and testing process can be found in Appendix 4.

A final UX evaluation of the fully developed application in Visual Studio Code was conducted together with the lecturer of the User Interface and Application Design course, who is also a professional UX designer. The session was held remotely through Microsoft Teams and focused on the application's visual design, usability, accessibility, interface consistency, color combinations, and overall UX/UI quality. Each view and functionality was reviewed in detail to identify remaining usability issues and areas for improvement.

The results revealed several recurring usability issues and opportunities for improvement. The main findings were as follows:

- Naming and structure confusion. Some features in the hi-fi prototype had unclear or misleading labels, which caused confusion among users. Participants suggested simplifying and renaming these elements to better reflect their purpose and functionality.
- Need for integrations and automation. Participants expressed a strong preference for integrating the system with existing tools, such as Google Calendar.

Additionally, automation features such as autofill and automatic schedule generation were highlighted as valuable enhancements that could improve efficiency.

- Colour contrast issues. Particularly among managers, there was a clear need for improved colour contrast to make important information more distinguishable and easier to access quickly.

### 4.4.2 Improvements

Based on the feedback collected from coursemates, service-industry workers, and a professional UX designer, several design and functionality improvements were implemented.

These improvements included simplifying navigation by reducing unnecessary pages, improving the clarity of labels and feature naming, and enhancing color contrast throughout the interface. Integration with Google Calendar was also added to improve workflow efficiency.

In addition, the shift-swapping process was improved by introducing clearer interaction logic and a more intuitive workflow. New features such as a notification button, an “Open Shifts” page, and expanded manager permissions and settings were also implemented to improve usability and make the application more customizable for different user needs.

Following the expert UX evaluation, additional improvements were made to refine color combinations, strengthen visual hierarchy, and make interface elements more consistent and intuitive. Particular attention was given to established UX design principles, accessibility, and usability guidelines in order to create a clearer and more user-friendly experience.

In conclusion, the ideas, prototypes, and functionalities of the application were continuously refined throughout the development process based on UX interviews, usability testing, and feedback sessions. These evaluations provided valuable insight into the real needs, expectations, and everyday challenges of end users working in the service industry. As a result, the development process became strongly user-centred, ensuring that the final solution was developed according to practical feedback and real-world usability needs rather than assumptions alone.

# 5 Application development and main functionalities

This chapter focuses on the development and validation of the Buka application. The primary objective was to create a unique, minimalistic, and modern solution that is intuitive, automated, simple, and time-efficient for both workers and company managers. Additionally, the project aimed to develop an automated scheduling system, which serves as the foundation of the frontend implementation.

## 5.1 Application goals and scope

The goal of Buka application was to mainly address inefficiencies in traditional work schedule management, which still often relies on manual tools or outdated platforms. Current solutions and approaches lead to miscommunication between coworkers, lack of transparency, a lot of manual work for managers and increased stress for all employees in the company. Additionally, many existing solutions do not provide a clear way to calculate wages, which can lead to mistakes and confusion among employees.

The scope of the system is to create a structured environment with three main user roles: company owners, managers, and employees. Each role is designed to have different levels of access and responsibility within the system. However, at the current stage of development, the company owner view has not yet been implemented. This means that while managers and employees can use the system effectively, higher-level administrative control is planned as a future improvement.

The system includes all core functionalities required for effective schedule management in service-industry companies, main functionalities being schedule creation and management, centralised storage and handling of employee data, handling of employee wages and bonuses, request-based workflows such as shift swaps, vacation requests, sick leave submissions, and preferred days off. The application also provides a user-friendly interface for both employees and managers. Additionally, users can connect their account with their Google account and Google Calendar, allowing them to view their work schedule alongside personal events and better manage their time.

## 5.2 Practical use and functionality of application

This chapter introduces the main pages and functionalities of the final application.

All figures are presented from the manager's view, as it includes the full set of features of the system. The employee view is largely similar in layout but offers fewer functionalities due to permission restrictions. Any differences between the two views are explained in the subsections.

### 5.2.1 Overview of the final solution

1. "Home" page.

As illustrated in Figure 20, the "Home" page serves as the entry point of the application, where users can familiarize themselves with the system, create an account, and log in.

A more detailed overview of the login and employee onboarding process is provided in Appendix 7.

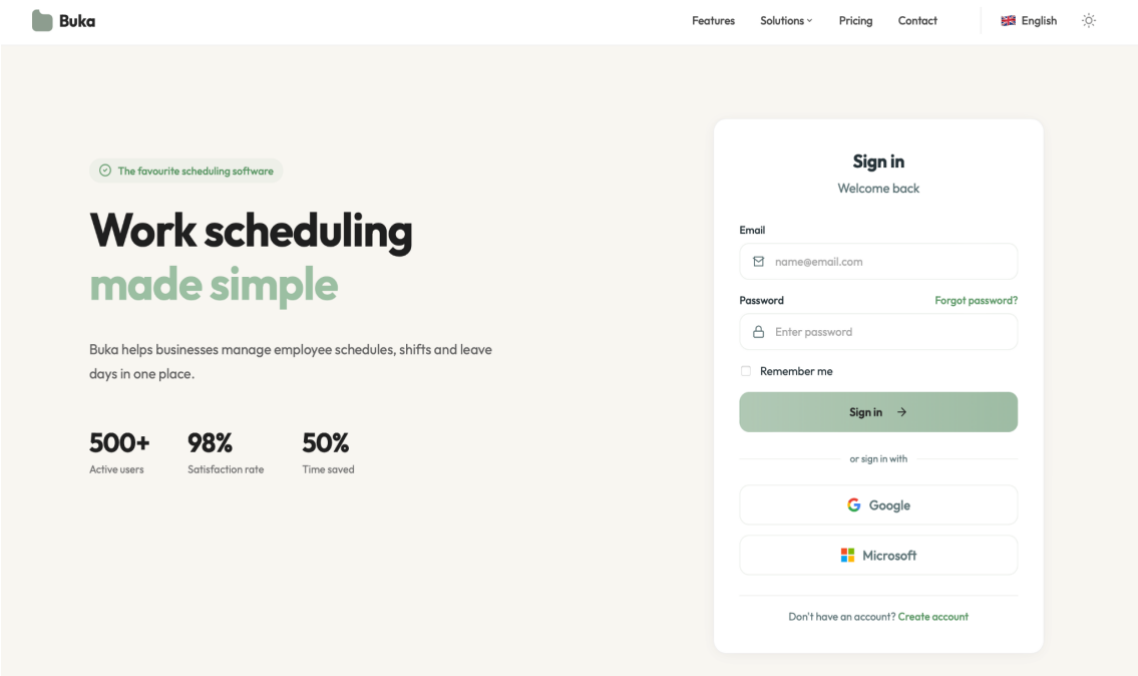


Figure 20. Buka's home page view.

2. "Shifts" (main) page.

As illustrated in Figure 21, the "Shifts" page represents the core functionality of the application. It provides a calendar view of all employees and allows users to view

different calendars and team members within the company. In the manager view, the page also displays sick leave, vacations, and days off in real time.

By interacting with the calendar, users can perform the main actions of the application (see sections 5.2.3 and 5.2.4). In the company manager's view, schedules can be autogenerated (see section 5.2.2), and company managers are able to view pending requests for sick leave, vacations, and days off.

Users can also customize the calendar view according to their preferences. As shown in Figure 22, users can adjust the size of the calendar cells, choose how many shifts to display per day, and modify other settings. Based on these preferences, the calendar view on the "Shifts" page updates accordingly. Figure 23 demonstrates a calendar view displaying only three shifts per day, while Figure 24 illustrated the calendar filtered to a single café location.

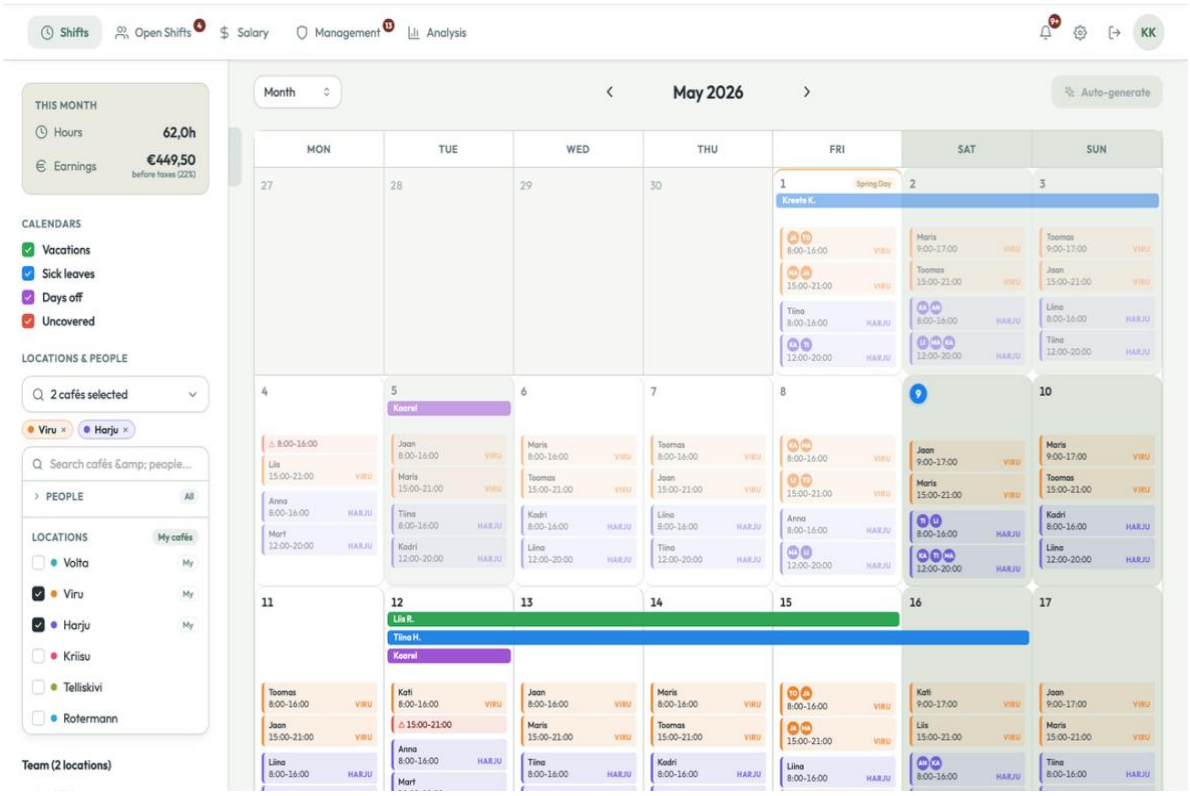


Figure 21. Buka's calendar view.

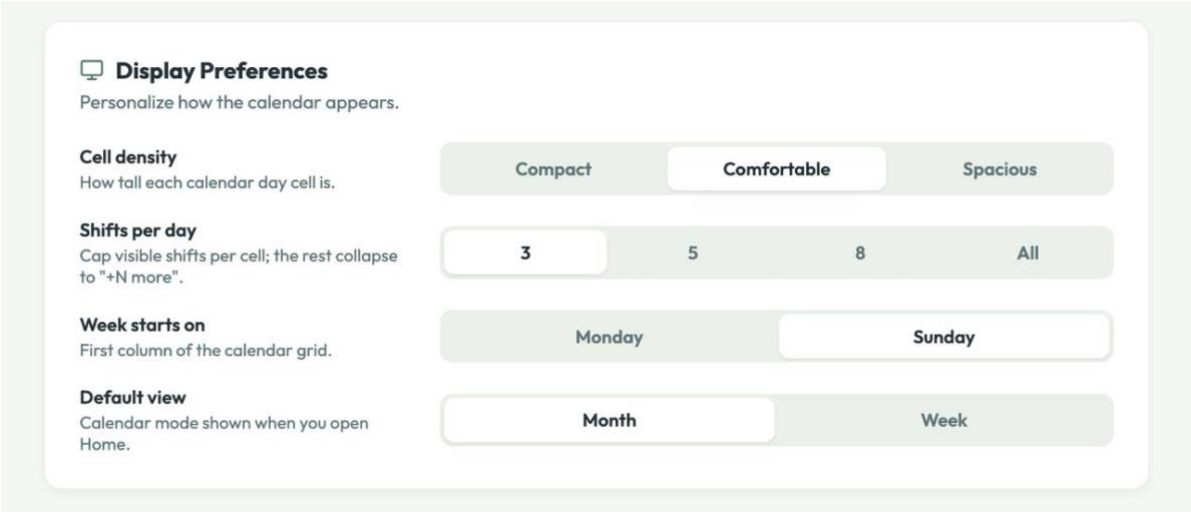


Figure 22. Calendar settings.

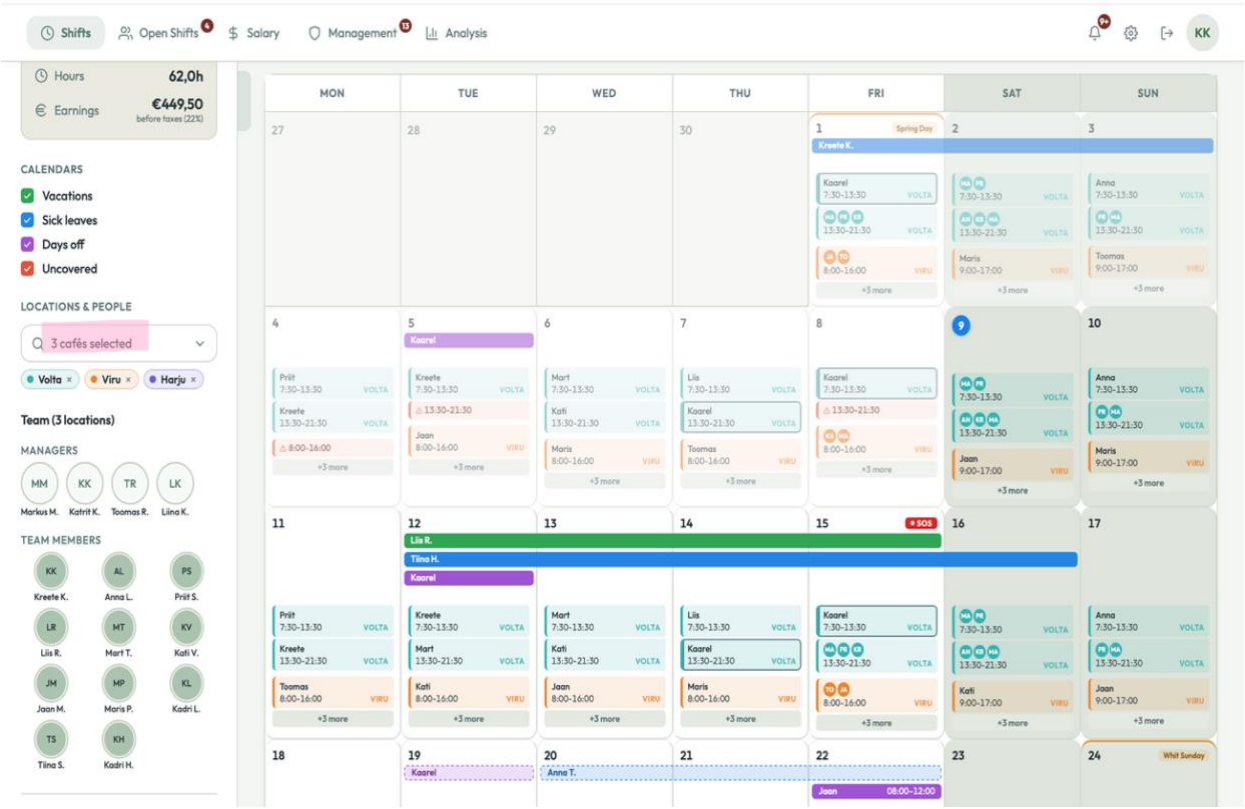


Figure 23. Calendar view displaying only three shifts per day.

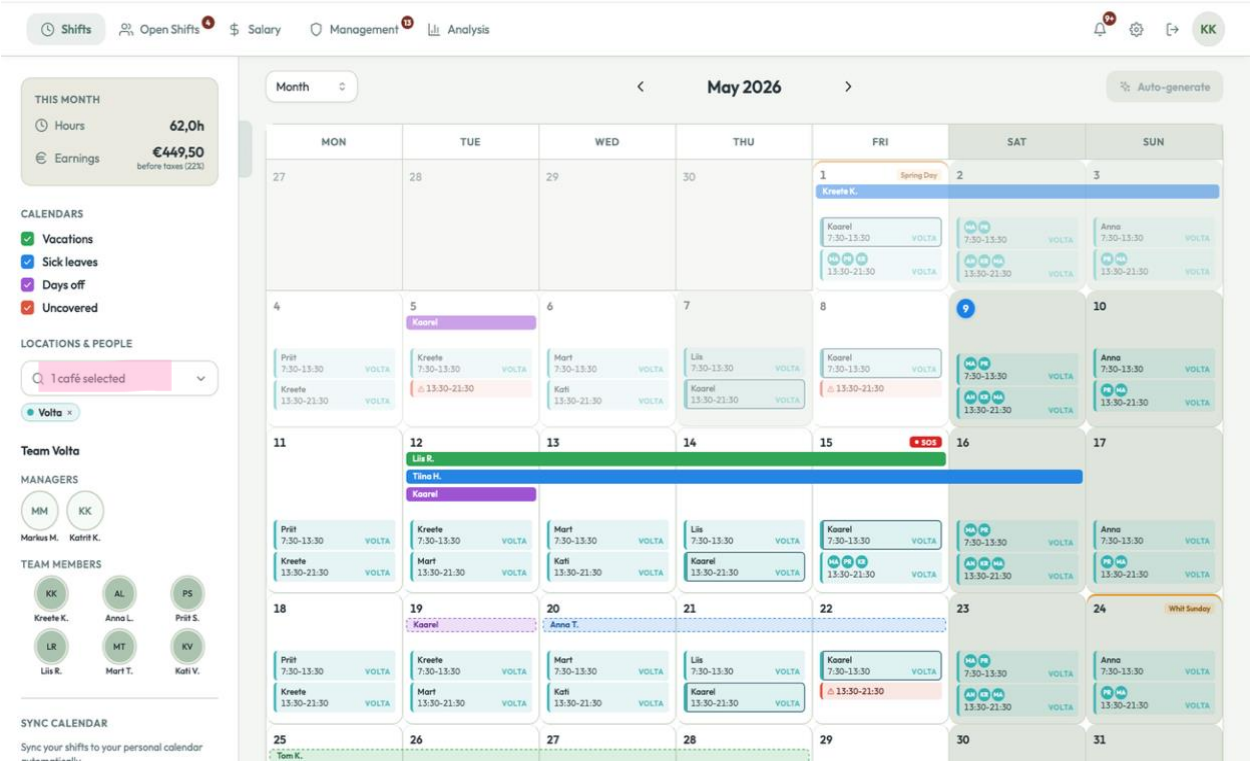


Figure 24. Calendar view filtered to a single cafe location.

3. “Open Shifts” page.

As illustrated in Figure 25, the “Open Shifts” page manages both available and SOS (urgent) shifts that users can claim.

Employees can submit shifts they are unable or unwilling to work, which are then made available to others based on predefined limits set by the manager (e.g., a specific percentage or number of shifts). This enables coworkers to claim additional shifts and earn extra income.

The SOS shifts section is designed for handling urgent situations, such as sick leave. When an employee reports sick leave, all their assigned shifts during that period are automatically transferred to the SOS section. Managers can view available employees and send notifications to quickly fill these shifts. Additionally, managers can manually select any critical shift from the calendar view and assign it to the SOS section.

This functionality provides a clear overview of shifts that require immediate attention and ensures efficient coverage of urgent shift coverage needs.

In the regular employee view, users cannot see “available employees” for SOS shifts or send notifications. They are only able to view and claim SOS shifts, as well as receive notifications regarding any related changes.

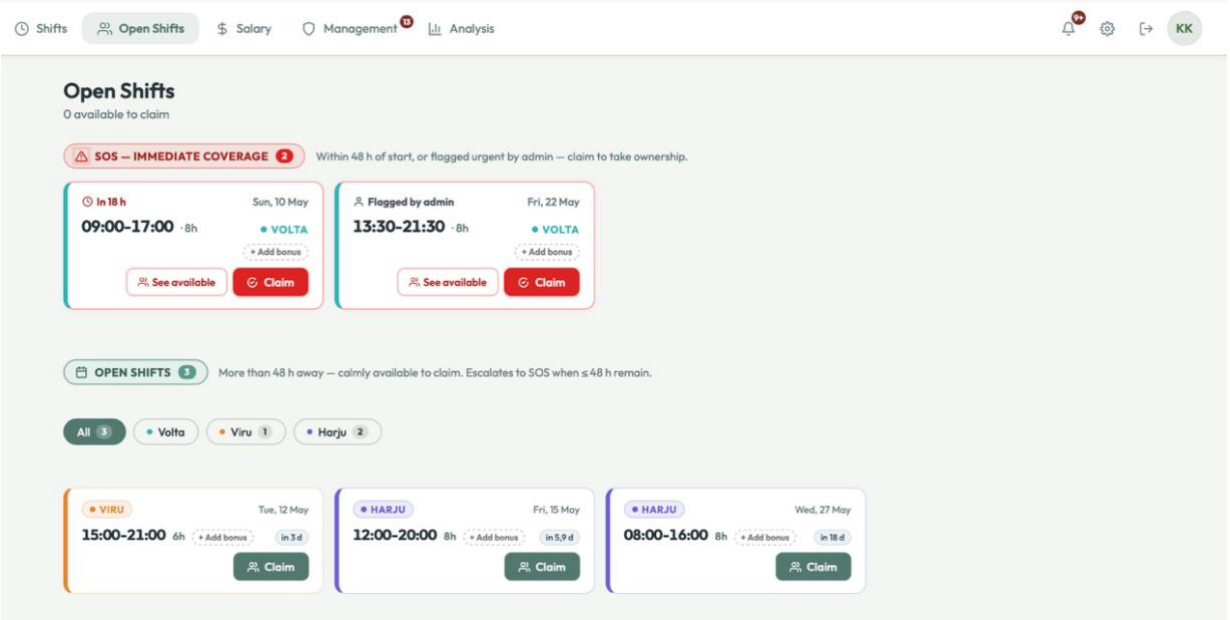


Figure 25. Open shifts and SOS shift management view.

4. “Salary” page.

As shown in Figure 26, the “Salary” overview page provides a summary of the user’s earnings, including salary, vacation, sick leave, and, where applicable, study leave details. The system calculates earnings in real time based on the user’s worked hours. More details about the wage-calculation logic can be found in Appendix 6.

The final salary figures become fully accurate at the end of the month, once the manager has added bonuses.

In the regular employee view, users do not have permission to modify salary settings or assign bonuses to other employees. These actions are restricted to the manager view.

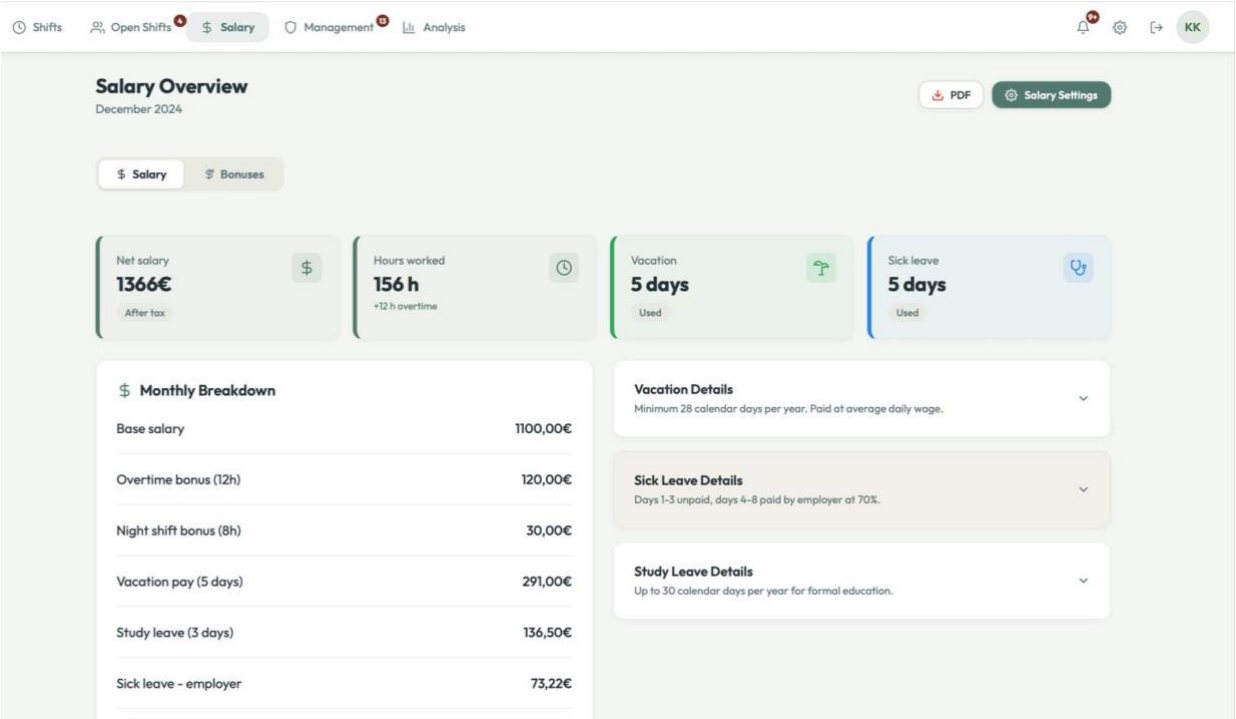


Figure 26. Salary overview page.

5. “Management” page.

As illustrated in Figure 27, the “Management” page includes all functionalities required for company and employee management.

The main features include the approval of join requests from new employees who wish to access the company’s system, as well as an overview of sent invitations. It also provides a section for managing shift swap requests, where company managers can review all requested shift changes submitted by employees.

The pending requests section displays employee leave, days off, and open shift requests, allowing managers to either approve or decline them.

The accepted history section provides an overview of all previously processed actions, including approvals and rejections, performed by the manager.

This view is only accessible in the company manager and cannot be accessed by regular employees.

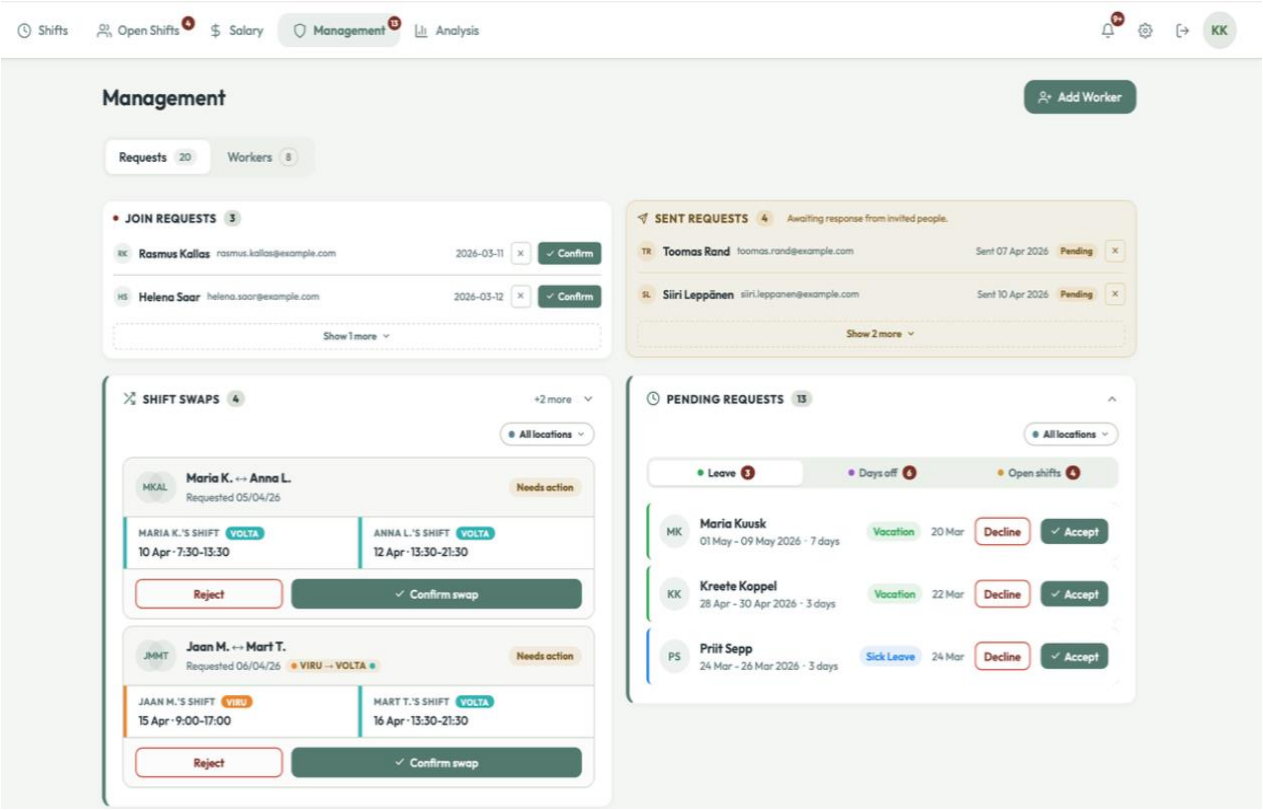


Figure 27. Management view.

### 5.2.2 Autogenerating schedule

As illustrated in Figure 28, the autogeneration functionality is one of the core features of the application. Using the autogenerate button, the manager can generate a draft schedule for the upcoming month.

Before generation, managers can configure generation preferences, including selected cafes, available employees, and worker absences, as shown in Figure 29. After the generation process is completed, the system provides an overview of the generated results, warnings and filled shifts, as illustrated in Figure 30.

The generated schedule is initially saved as a draft view, allowing managers to review and modify shifts before finalising and publishing the schedule, as illustrated in Figure 31. More details about the autogenerating logic are provided in Appendix 5.

This functionality is available only in the company manager view. Regular employees can view the schedule only after it has been approved and published by the manager.

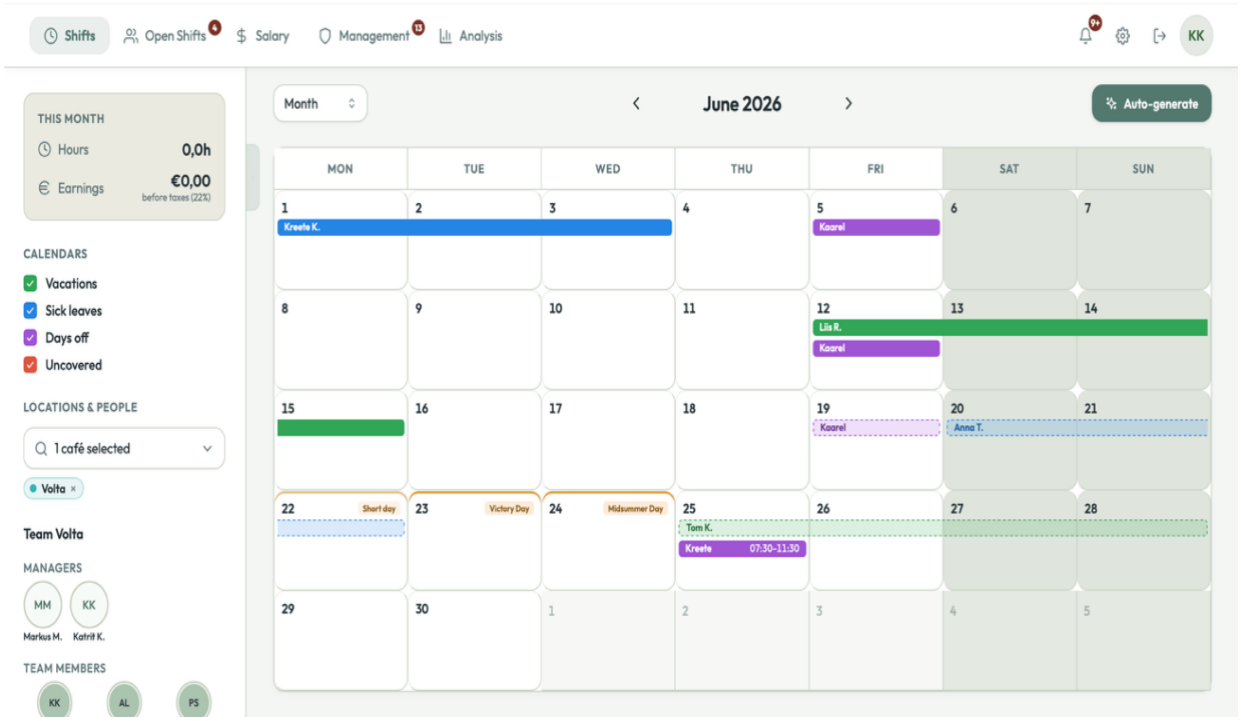


Figure 28. Calendar view before schedule generation.

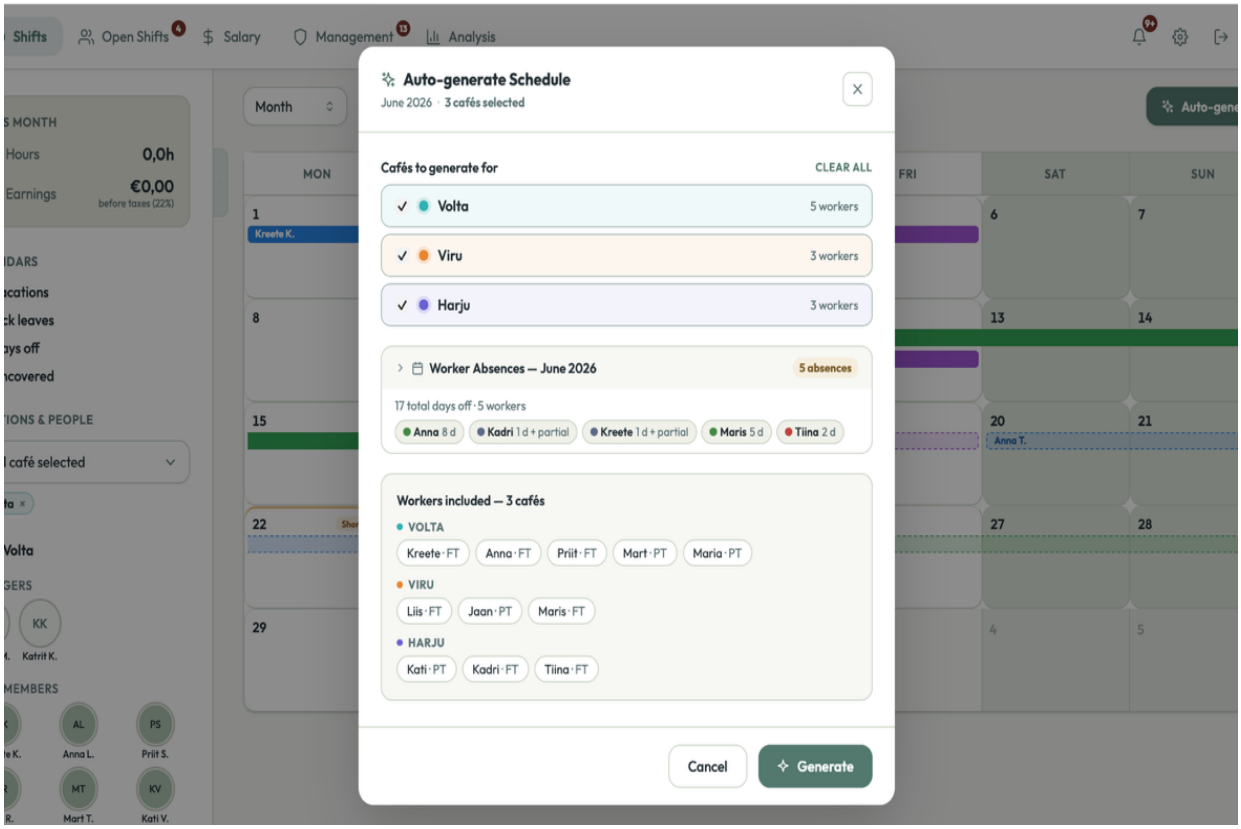


Figure 29. Autogeneration settings.

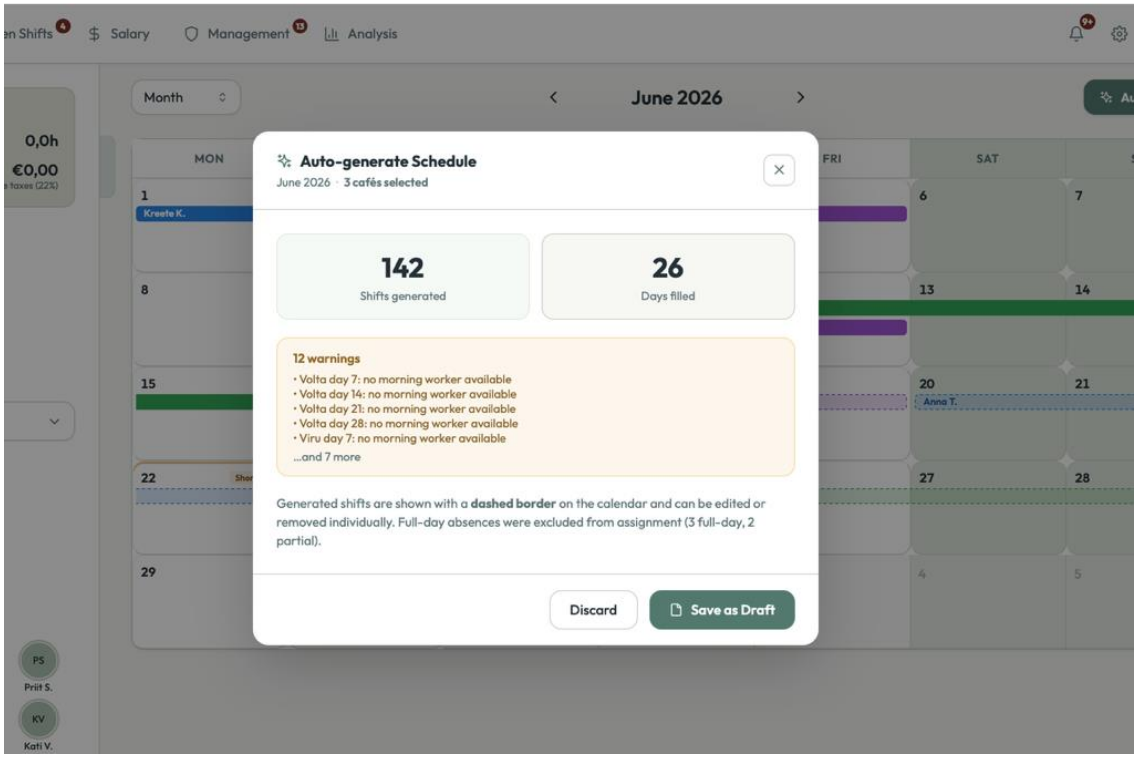


Figure 30. Autogeneration results.

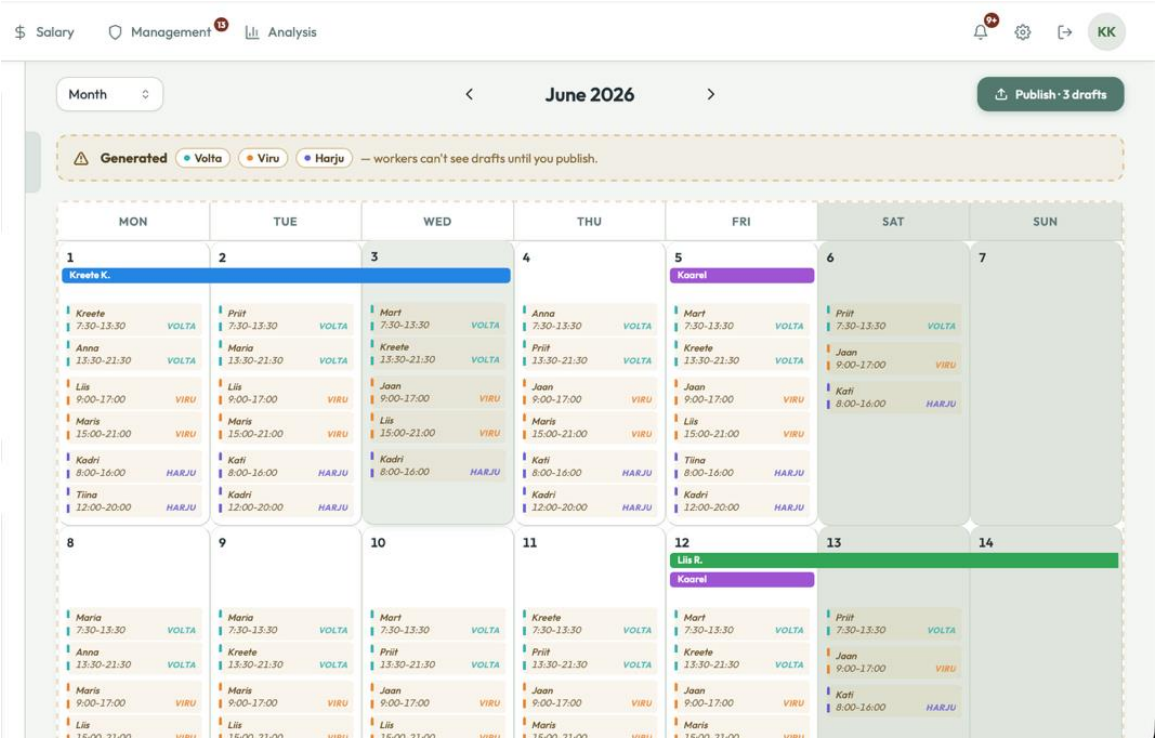


Figure 31. Generated schedule draft view.

### 5.2.3 Shift swapping

As illustrated in Figures 32, 33 and 34, shift swapping is one of the most common needs in the service sector, yet traditionally one of the most inefficient processes. An employee giving up a shift must contact colleagues individually, wait for responses, and then obtain manager approval, all through unstructured personal messages. The process is time-consuming and prone to communication errors and forgotten requests.

Buka addresses this issue by integrating shift swapping directly into the calendar view. Figure 32 demonstrates the shift details and actions interface, where users can access shift-related actions directly from the calendar. Employees can either send a swap request to a specific coworker, as shown in Figure 33, or publish it to the "Open Shifts" section, illustrated in Figure 34, where any eligible colleague can claim it.

All requests reach the manager for final approval, ensuring compliance with internal rules while removing the need for informal communication.

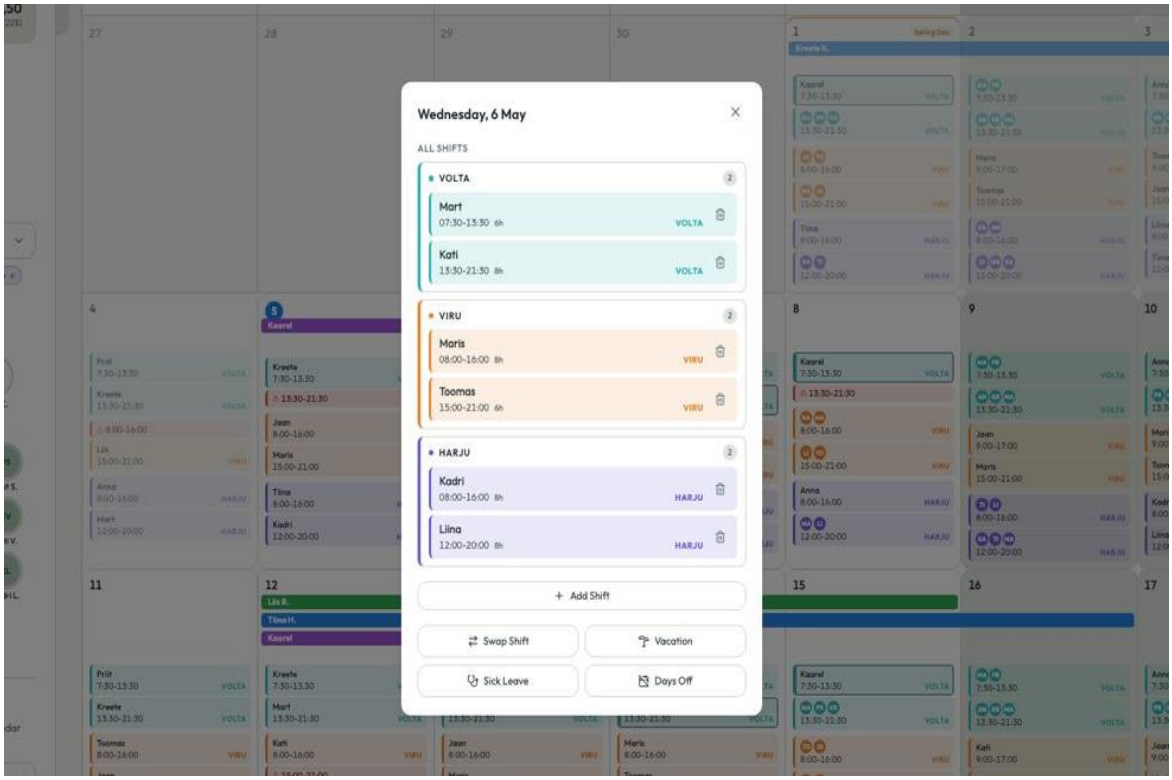


Figure 32. Shift details and actions.

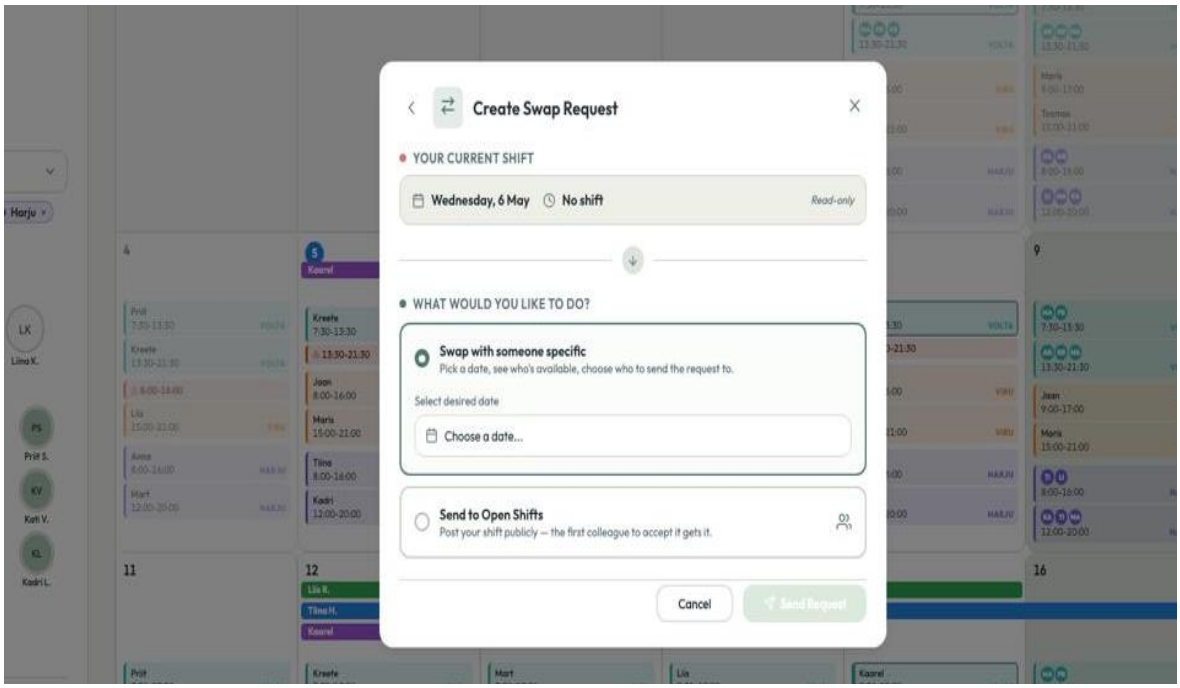


Figure 33. Creating a swap request (specific user selection).

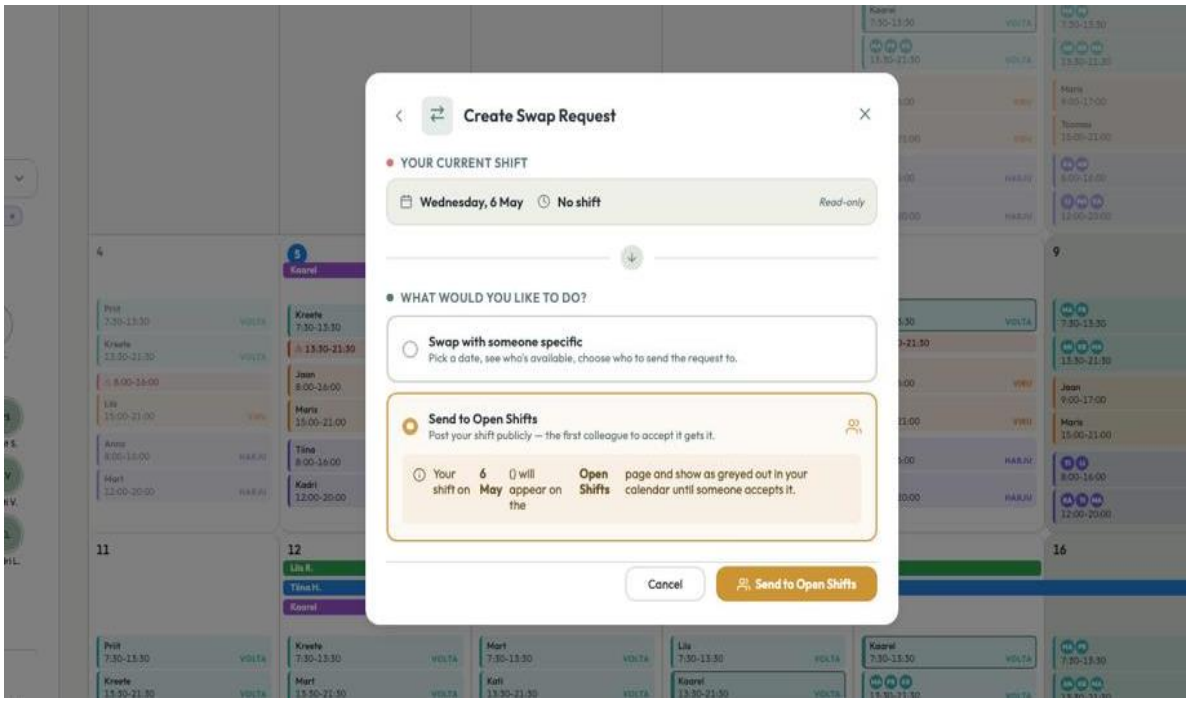


Figure 34. Creating a swap request (sending shift to open shifts).

### 5.2.4 Submitting days off, vacations, sick leaves

This view is identical for both administrators and regular employees. Users can submit vacation requests, sick leaves, and days off directly through the main calendar view.

Sick leaves and days off can only be submitted for the current month, as shown in Figures 35 and 36, whereas vacation requests can also be submitted for upcoming months, as illustrated in Figure 37. Both vacations and days off are subject to monthly deadlines, by which all employees must submit their requests. If the deadline is missed, the submission option is no longer available. In contrast, sick leave can be submitted at any time during the month.

Once a user submits a request, the company manager receives notifications via the notification bell and on the “Management” page. In addition, submitted requests are visually represented in the calendar view with a dashed line. After the manager approves the request, the dashed line changes to a solid colour, indicating confirmation. This allows the manager to instantly see all approved and pending requests in the calendar, providing a clear overview of employee availability and workforce planning for the upcoming month.

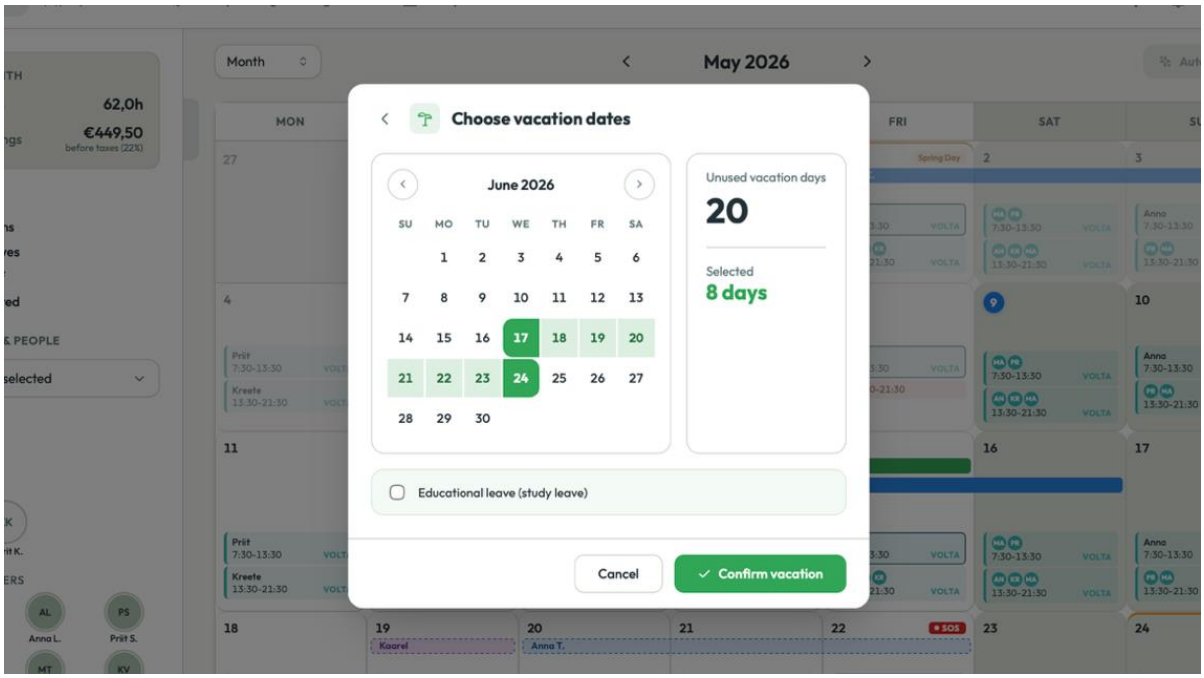


Figure 35. Submission of vacation days.

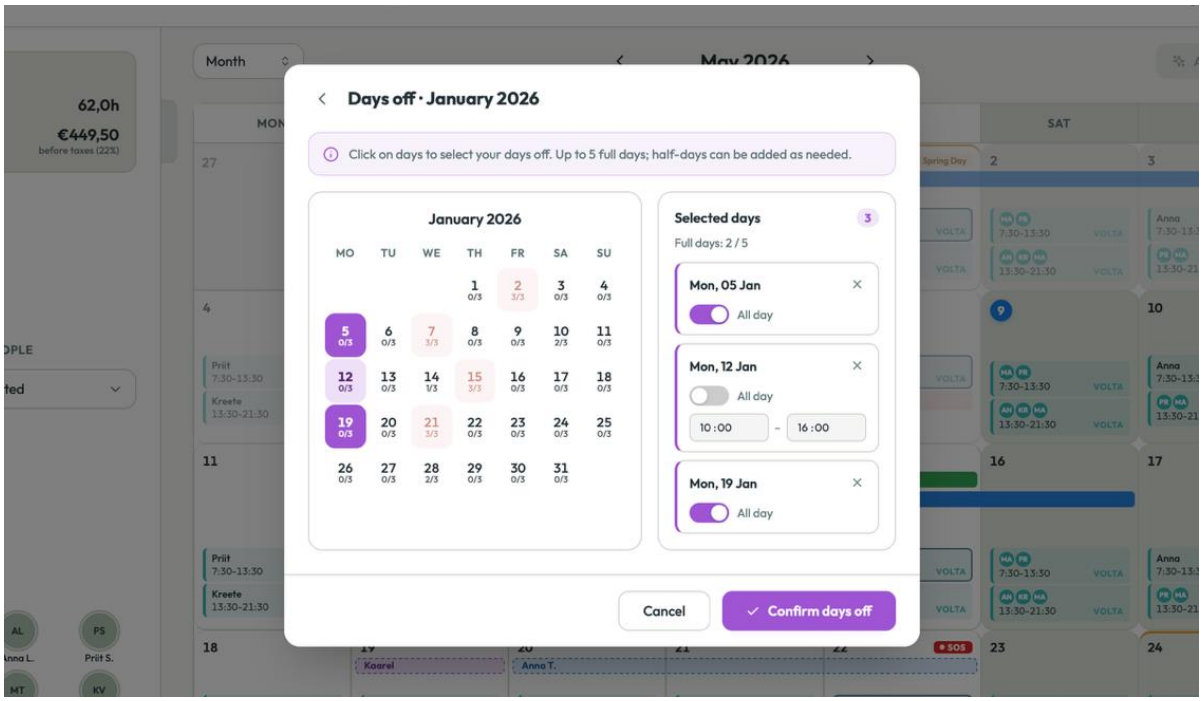


Figure 36. Submission of days off.

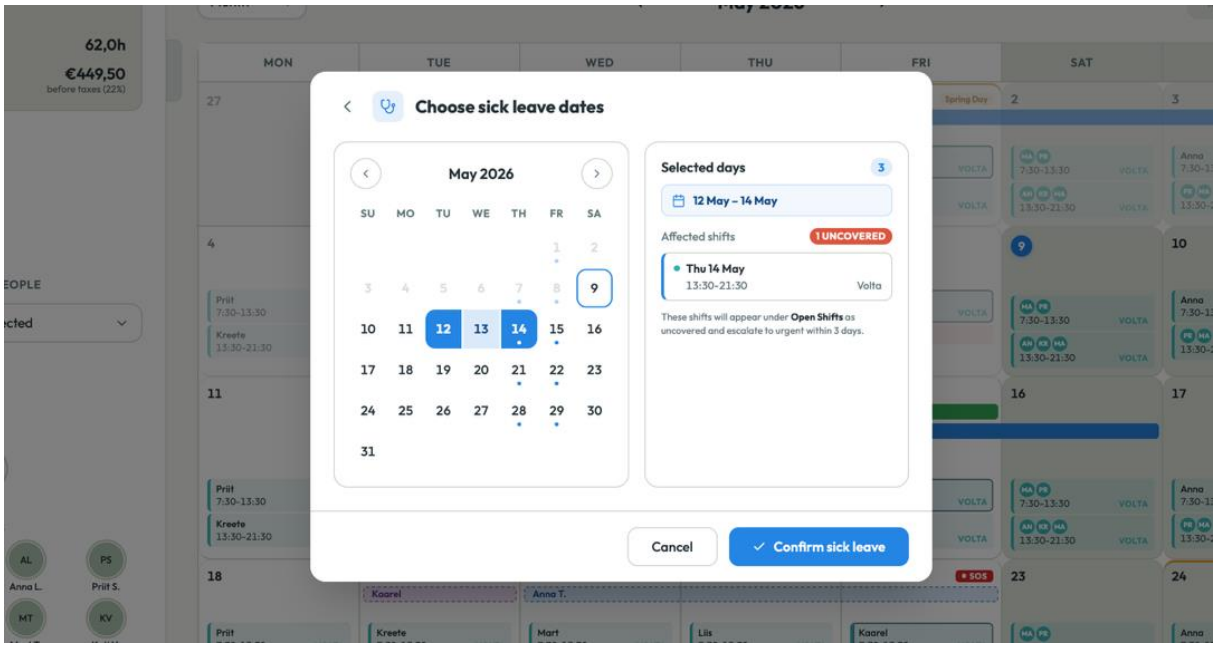


Figure 37. Submission of sick leaves.

### 5.2.5 Submitting of bonuses

On the “Salary” page, the company manager can assign bonuses to employees based on their performance. Bonuses can be defined either as a fixed hourly addition or as a monthly amount. Figure 38 demonstrates the bonus assignment form, where managers can specify the bonus type, amount, and an optional note.

Before submission, the manager is required to confirm the bonus assignment, as shown in Figure 39. After confirmation, the updated bonus information is displayed in the employee bonus overview, illustrated Figure 40. The assigned bonuses are also reflected on each employee's salary overview page, where both pre-tax and post-tax values are shown.

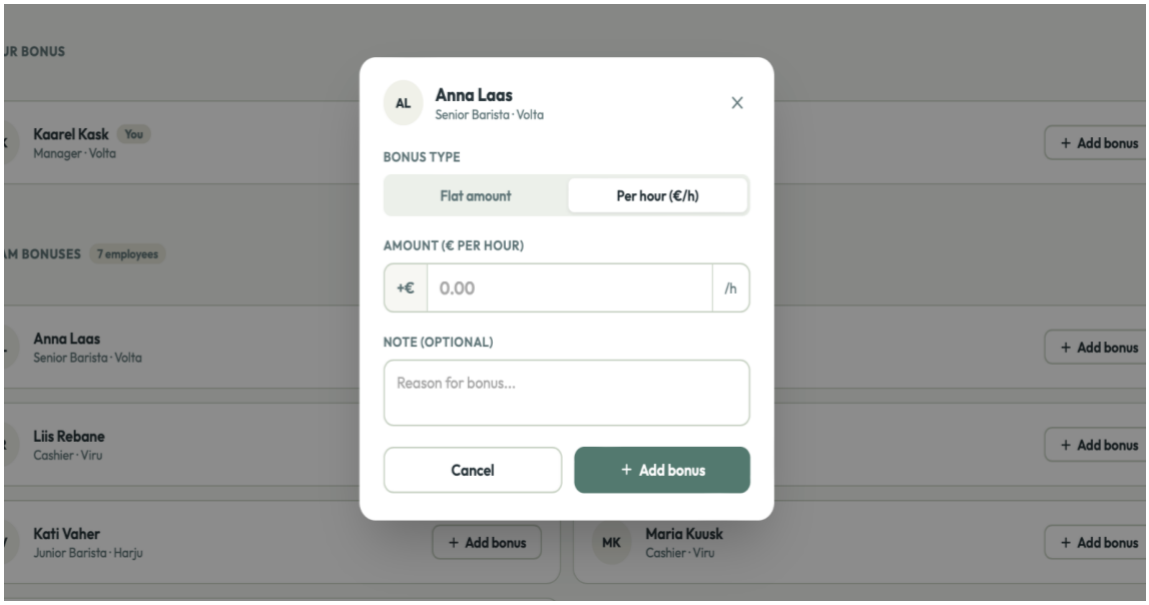


Figure 38. Bonus assignment form.

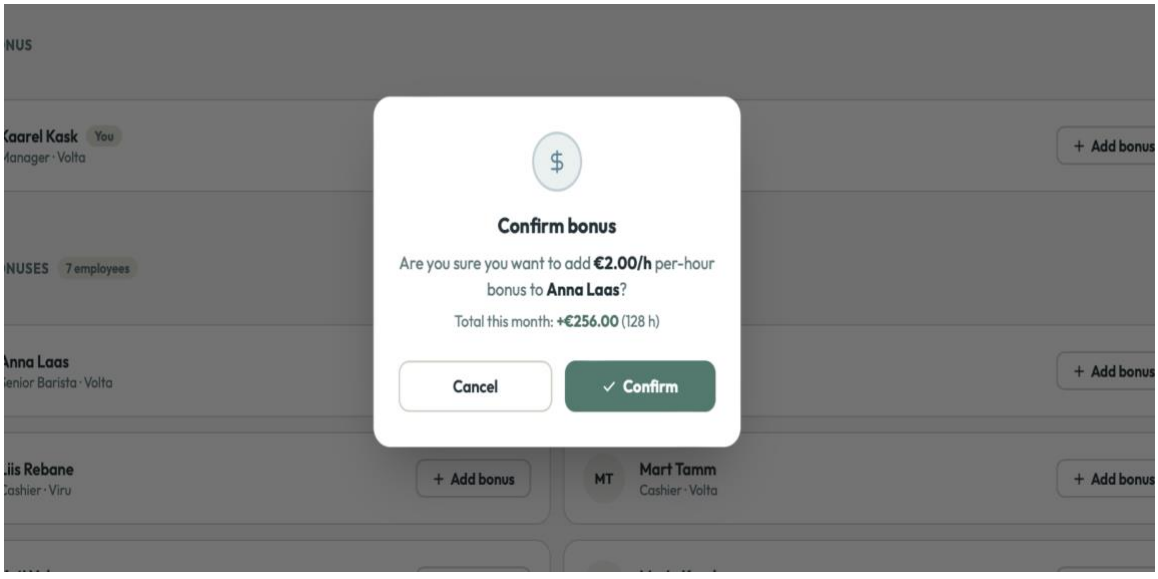


Figure 39. Bonus confirmation dialog.

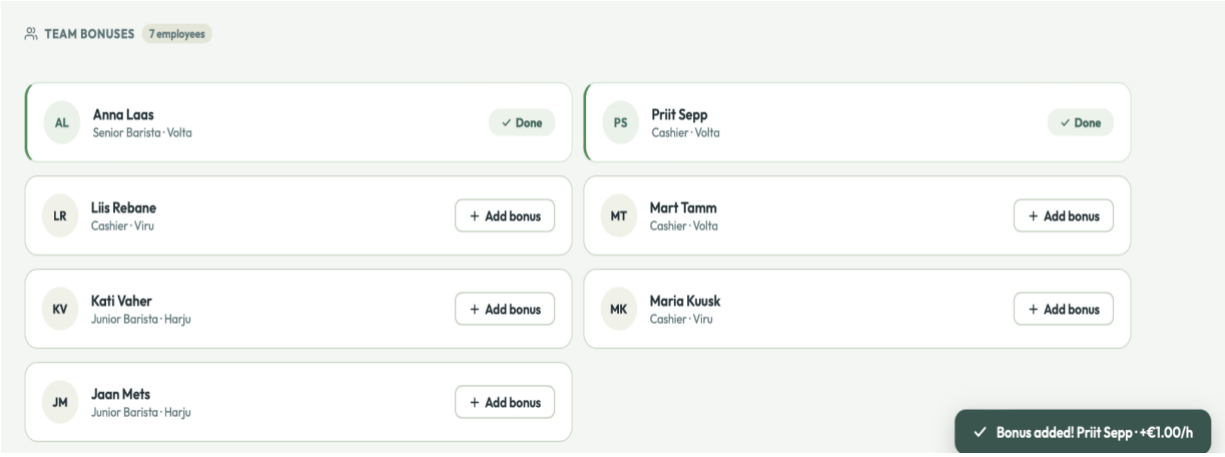


Figure 40. Updated bonus list after submission.

### 5.2.6 Notifications

The application includes a notification system, accessible through a notification bell icon, which is designed to keep users informed about relevant updates and actions within the system, as illustrated in Figure 41.

The company manager receives a set of notifications, which include deadline reminders, join requests, shift swap requests, leave requests, days-off requests, shift claims, shift drops, profile changes, and document uploads. Notifications provide managers with a clear overview and enable them to respond quickly to any changes.

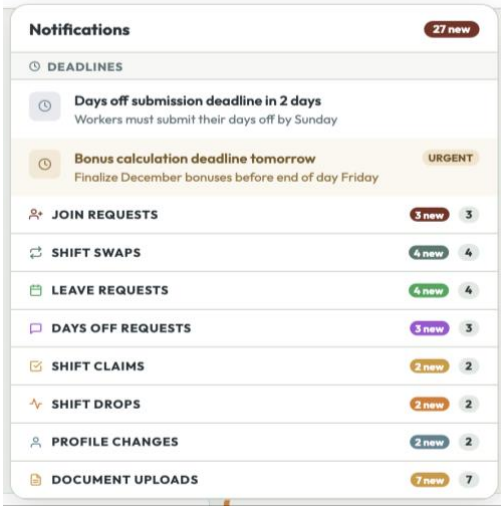


Figure 41. Notification view for company manager.

In contrast, employees receive a more limited and focused set of notifications. They are notified only about relevant deadlines, shift swap requests, changes to their shifts, SOS shifts and open shifts.

# 6 Analysis and conclusions

This chapter presents an analysis of the project results and evaluates how effectively the developed solution achieved its objectives. It begins with an overall assessment of the application and continues with an analysis of teamwork and collaboration throughout the development process. The chapter also examines the market positioning and competitive value of the application, followed by an overview of the proposed business plan and the future potential of the solution.

In addition, the chapter reflects on the key lessons learned during the project, including both technical and organisational insights gained by the authors. Finally, possible future development directions are discussed, focusing on feature expansion, scalability, and the potential of further developing Buka into a start-up product.

## 6.1 Overall assessment

The primary goal of this thesis was to design and validate a user-centred digital workforce management solution, with a special focus on the needs of the Estonian service sector. Based on the outcomes of the project, it can be said that all the objectives set at the beginning, were successfully achieved.

The first objective, improving workplace transparency and employee autonomy, was achieved even better than initially expected. This was mainly due to continuous validation with end-users throughout the development process (as described in Chapter 4.4), as well as collaboration with a professional UX designer. These factors ensured that the solution stayed closely aligned with real user needs, resulting in a system that provides clear visibility into schedules, leave requests, and salary-related information.

The second objective, reducing administrative workload and improving accuracy, was also successfully achieved. The developed solution replaces fragmented manual processes with structured digital workflows, which significantly reduces the likelihood of human error and improves efficiency in scheduling, leave management, and

communication. As a result, the system reduces the workload for managers while still ensuring smooth workflow and meeting organisational and legal requirements.

During the development process, several additional ideas for future improvements were identified (more in Chapter 6.6). All ideas for future development show both the flexibility of the solution and its potential for further development beyond the scope of this thesis.

Overall, the developed system meets all the requirements set for the project and successfully fulfils its intended purpose. It provides a practical, scalable, and user-friendly alternative to other scheduling methods, helping to improve transparency, efficiency, and work-life balance in the service sector.

## 6.2 Teamwork analysis

Table 2 provides a consensual overview of the contribution of each team member throughout the project.

Table 2. Overview of team contributions.

| **Task** | **Demi Heinsoo** | **Kreete Keres** |
|---|---|---|
| Competitor analysis | -1 | +1 |
| User flows | +1 | -1 |
| Wireframing | 0 | 0 |
| Lo-fi prototypes | -1 | +1 |
| Hi-fi prototypes | +1 | -1 |
| UX interviews | 0 | 0 |
| User testing | 0 | 0 |
| Frontend discussions | 0 | 0 |
| Application development | +1 | -1 |
| Document writing | -1 | +1 |
| Meetings with supervisors | 0 | 0 |
| Project planning | 0 | 0 |
| **Total** | **0** | **0** |

The evaluation is based on the agreed assessment between team members, where +2 indicates a significantly larger contribution, +1 indicates a larger contribution, 0 indicates equal contribution, -1 indicates a smaller contribution, and -2 indicates a significantly smaller contribution within a specific task.

Overall, the teamwork throughout the Buka project was balanced and collaborative. Both team members contributed equally to the overall development process, as reflected by the final total assessment score of 0 for each member. While responsibilities naturally varied depending on the task, the workload remained evenly distributed throughout the whole project.

The teamwork process was supported by continuous communication, regular discussions, and shared decision-making throughout the project. This collaborative approach enabled efficient problem-solving and ensured that both technical and design-related decisions were aligned with the project goals.

The total workload amounted to approximately 16 ECTS, equivalent to 416 hours. This included the User Interface and Application Design course (6 ECTS), competitor analysis, meetings with supervisors and team members, application development, UX interviews, usability testing, prototype iterations, and thesis writing.

Unfortunately, detailed work logs for the entire process are not available, since the project was started already during the autumn semester before the official thesis process had begun. However, parts of the development process are visible through Git commit history and the different prototype and application iterations.

## 6.3 Market positioning and competitive evaluation of the application

This section evaluates Buka's position within the current workforce management software market, with a special focus on functionality and user experience. The evaluation is based on the competitor analysis presented in Chapter 3.1, as well as insights gathered during user interviews, testing, and application development.

Compared to existing platforms such as Begin, Deputy, Wemply, and RotaCloud, Buka demonstrates several competitive advantages, particularly in usability, flexibility, and modern user experience design. While Begin and Wemply offer extensive functionality and strong localization for the Estonian market, both platforms are often perceived as overly complex and visually outdated. Buka addresses these issues by providing a simplified and intuitive interface that allows users to perform scheduling-related tasks with minimal effort.

Deputy and RotaCloud provide strong mobile accessibility and modern interfaces, especially for employees. However, these systems may rely heavily on third-party integrations and may not fully support the operational needs of Estonian service-industry companies. Buka aims to bridge this gap by combining practical functionality with a modern and user-friendly experience.

One of Buka's key features is its schedule-generating algorithm, which automatically creates schedules based on employee availability, preferences, workload balancing, and manager-defined rules. Unlike many existing systems that still require considerable manual work, Buka reduces administrative workload and improves scheduling efficiency. Another important feature is the transparent real-time wage calculation system, which allows employees to continuously track their worked hours, bonuses, and estimated salary directly within the application.

Although Buka does not yet include all planned features, the application has already demonstrated strong potential within the small and medium-sized business market. User testing confirmed that the platform successfully addresses several common issues found in existing scheduling solutions, particularly regarding usability, accessibility, and efficiency.

The analysis confirms that Buka is competitively positioned within the workforce management software market. By combining intelligent automation, simplified workflows, and a modern user experience, the application provides a practical alternative to both traditional scheduling methods and existing workforce management platforms used in the Estonian service industry.

## 6.4 Business plan

As illustrated in Figure 42, Buka is planned to operate using a Software-as-a-Service (SaaS) subscription model, where pricing is determined on a per-user-per-month basis. The pricing structure is designed to scale according to the size of the company and the level of functionality and integrations required by the client. The planned pricing model is also influenced by the pricing strategies of existing competitors within the workforce scheduling and management software market.

- **Package 1** is planned to cost €6.5 per user per month and includes all core functionalities currently available in the application. These functionalities include automatic schedule generation, real-time wage calculation, calendar synchronization with platforms such as Google Calendar, notifications, shift swaps, open shifts, and the submission of vacations, sick leaves, and days off requests. This package is designed to provide small and medium-sized businesses with a complete and user-friendly workforce management solution that reduces manual scheduling work and improves overall internal communication.
- **Package 2** is planned to cost €8 per user per month and includes all features from Package 1 together with integrations with widely used Estonian accounting and payroll software solutions. Planned integrations include systems such as Merit Aktiva [37], Merit Palk [38], and SimplBooks [39]. These integrations are intended to simplify payroll management, reduce duplicate administrative tasks, and improve the efficiency of both financial and workforce-related processes within companies.
- **Package 3** is planned to cost €10 per user per month and includes all previous functionalities together with planned AI-supported features. These future functionalities are intended to further reduce the workload of managers and administrators by providing intelligent schedule recommendations, workforce optimization suggestions, predictive staffing analysis, and AI-assisted automation tools. The goal of these features is to improve decision-making, speed up managerial workflows, and increase overall operational efficiency.

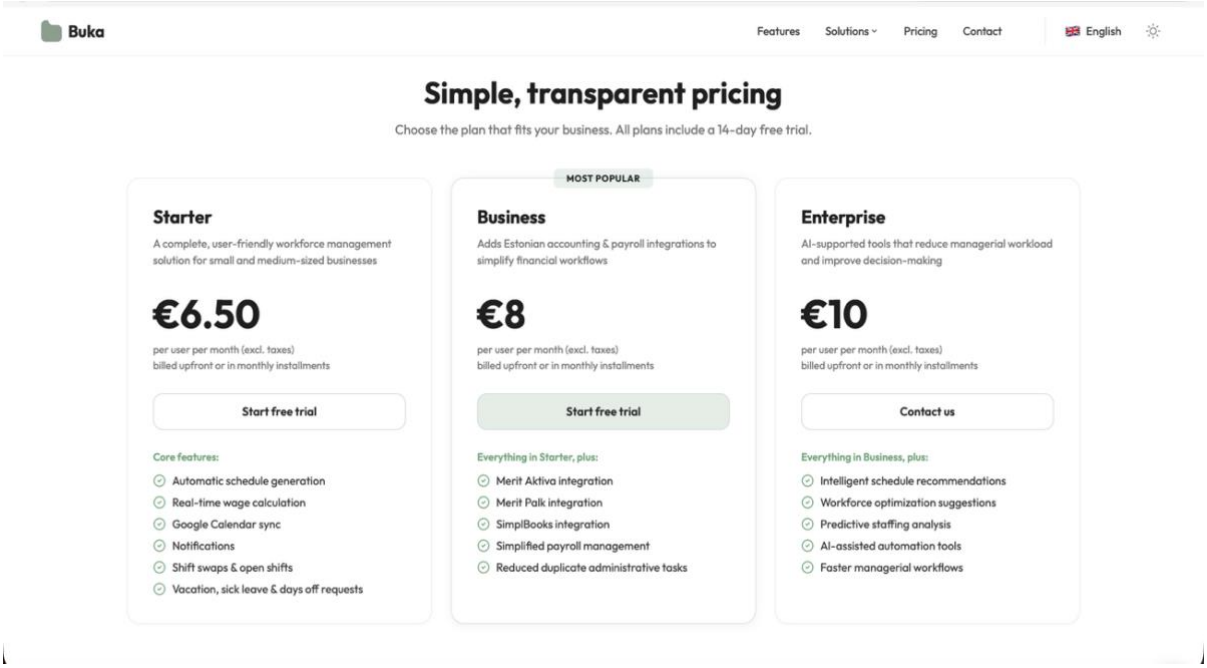


Figure 42. Buka's pricing list.

The primary target market for Buka consists of organisations with shift-based workforces, including restaurants, cafés, hotels, retail businesses, healthcare institutions, and other service-sector companies. Since the application was developed based on real-world scheduling challenges experienced in the hospitality industry, the platform is particularly well suited for dynamic work environments, where schedules frequently change and efficient communication between employees and managers is essential.

The main strategy for building the initial client base is to focus on direct outreach, networking, and collaborations with local businesses in the service industry. Early user feedback and real-world implementation are essential for validating the business model, identifying improvement opportunities, and adapting the platform to practical workforce management needs. Once an initial client base has been established, acquiring new customers is expected to become significantly easier through client feedback, recommendations, references, and real-world success stories. The long-term goal is to establish Buka as a modern and intelligent workforce management platform that simplifies scheduling, improves payroll transparency, reduces administrative workload through automation, and integrates multiple workforce-related processes into one centralised system.

## 6.5 Lessons learned

The development process of Buka provided several important lessons, particularly in the areas of UX/UI design, teamwork, communication, and project planning.

One of the main takeaways was understanding the importance of continuously testing solutions with real users throughout all stages of the development process, including low-fidelity prototypes, high-fidelity prototypes, and the final solution. Feedback gathered from people working in the service industry, participation in a User Interface and Application Design course, and consultations with a professional UX designer provided valuable insights into everyday scheduling challenges and helped identify which functionalities and workflows required improvement. These experiences also improved the understanding of core UX/UI principles and best practices.

Another important lesson was related to communication and collaboration between frontend and backend development. Since the solution was developed as two separate thesis projects, maintaining clear communication and a shared understanding between both sides of the system became essential. Regular discussions regarding technical requirements, user needs, and implementation details helped ensure that the final solution remained technically functional while also providing a smooth and intuitive user experience.

The project also highlighted the importance of proper planning and maintaining consistent effort throughout the entire development process. Although work on the thesis began relatively early during the User Interface and Application Design course in autumn, more consistent effort from the beginning would have reduced the workload and time pressure experienced near the final deadlines. This demonstrated the importance of long-term planning, task prioritization, and maintaining steady progress in larger software development projects.

In addition, the development process emphasized the importance of flexibility and iterative improvement. Several design and functionality decisions evolved throughout development as we gathered new insights from user testing and practical implementation. This clearly demonstrated that successful product development requires continuous work and adaptation based on real user needs, rather than relying on initial assumptions.

## 6.6 Future development

One of the main priorities for future development is improving the company onboarding and authentication system. While the current solution is functional, the focus during this thesis was on developing a working scheduling system and validating its core functionalities. Future work will therefore aim to create a more scalable, intuitive, and efficient company setup and login system suitable for organisations with more complex structures.

Future development also includes AI-supported functionalities to simplify administrative tasks. The planned integration would allow managers to interact with the system using natural language queries related to employees, schedules, payroll, and workforce management. It could additionally assist with operational tasks such as automatically notifying available employees about uncovered shifts or suggesting suitable replacements based on availability and workload, thereby reducing manual effort and supporting smarter decision-making.

Another planned addition is a dedicated "Owner" role for company owners and upper-level management. This role would offer broader organisational control, including the ability to define permissions for employee and manager views, approve manager-assigned bonuses, and monitor managerial activities. It would also provide analytical and financial overviews covering payroll costs, overtime and vacation expenses, workforce efficiency, and income and expense reporting.

Finally, a dedicated mobile application is a key future development goal. During this thesis, the focus was placed on the web-based application, and a fully developed mobile version was not completed due to time constraints. However, mobile accessibility is highly important for service-sector employees and managers who require convenient access to schedules, notifications, shift management, and communication features while on the move. Future work will therefore focus on developing native or cross-platform mobile applications to deliver a seamless experience across devices.

# 7 Summary

This thesis focused on researching, designing, developing, and validating a user-centred workforce management solution for the Estonian service sector. Despite the availability of digital tools, many cafés, restaurants, and seasonal businesses still rely on spreadsheets, paper schedules, and informal messaging, which often leads to inefficiency and communication issues. Buka was created to address these problems.

The project followed an iterative and user-centred development process that included competitor analysis, qualitative research, UX/UI design, and Scrum-based development. Existing platforms such as Begin, Deputy, Wemply, and RotaCloud were analysed to identify their strengths and weaknesses. User personas, lo-fi and hi-fi prototypes, UX interviews, and usability testing with employees, managers, and a professional UX designer were used throughout the process to improve the application based on real user feedback.

The final application includes key workforce management features such as automatic schedule generation, shift swapping, open and SOS shifts, leave requests, real-time wage calculation, customizable calendar views, bonus management, and notifications. These features help managers reduce administrative work while giving employees better visibility and control over their schedules and requests.

The results of the project show that the main objectives were successfully achieved. Buka improves transparency, communication, and efficiency within workforce management while providing a simpler and more modern alternative to both manual scheduling methods and existing solutions. Although the current version represents a strong MVP, several future development opportunities were identified, including AI-supported tools, deeper payroll integrations, an expanded owner role, and a dedicated mobile application.

# References


[1] The Gap Inc., “Stable Scheduling Study: Health Outcomes, Family Life, and Design Principles,” worklifelaw.org, 2018. [Online]. Available: https://worklifelaw.org/publications/Stable-Scheduling-Study-Report.pdf. [Accessed 8 April 2026].

[2] Deloitte, “2025 Gen Z and Millennial Survey,” deloitte.com, 2025. [Online]. Available: https://www.deloitte.com/content/dam/assets-shared/docs/campaigns/2025/2025-genz-millennial-survey.pdf. [Accessed 8 April 2026].

[3] Riigi Teataja, “Palgaseadus,” riigiteataja.ee, 2009. [Online]. Available: https://www.riigiteataja.ee/akt/103022026015. [Accessed 5 May 2026].

[4] J. W. Creswell, Research Design: Qualitative, Quantitative, and Mixed Methods Approaches, Thousand Oaks, CA: SAGE Publications, 2018.

[5] User Interviews, “How to Conduct User Interviews,” userinterviews.com, [Online]. Available: https://www.userinterviews.com/ux-research-field-guide-chapter/user-interviews. [Accessed 30 April 2026].

[6] Microsoft, “Microsoft Teams,” teams.live.com, [Online]. Available: https://teams.live.com. [Accessed 11 April 2026].

[7] J. Nielsen and T. K. Landauer, “A mathematical model of the finding of usability problems,” in *Proc. ACM INTERCHI’93 Conf. Human Factors in Computing Systems*, Amsterdam, 1993.

[8] Microsoft, “Visual Studio Code,” code.visualstudio.com, [Online]. Available: https://code.visualstudio.com. [Accessed 3 April 2026].

[9] GitLab Inc., “GitLab,” about.gitlab.com, [Online]. Available: https://about.gitlab.com. [Accessed 11 April 2026].

[10] Figma Inc., “Figma,” figma.com, [Online]. Available: https://www.figma.com. [Accessed 11 April 2026].

[11] Lovable Technologies, “Lovable,” lovable.dev, [Online]. Available: https://www.lovable.dev. [Accessed 11 April 2026].

[12] Google LLC, “Google Drive,” drive.google.com, [Online]. Available: https://drive.google.com. [Accessed 11 April 2026].

[13] OpenAI, “ChatGPT (2025) [Suur keelemudel],” chatgpt.com, 2025. [Online]. Available: https://chatgpt.com. [Accessed 11 April 2026].

[14] Anthropic, “Claude (2025) [Suur keelemudel],” claude.ai, 2025. [Online]. Available: https://claude.ai. [Accessed 11 April 2026].

[15] Wikipedia, “Scrum (project management),” en.wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/Scrum_(project_management). [Accessed 12 April 2026].

[16] Pixetic, “What is UI? What is UX? What’s the difference?,” pixetic.com, [Online]. Available: https://pixetic.com/blog/what-is-ui-what-is-ux-whats-the-difference/. [Accessed 13 April 2026].

[17] Wikipedia, “User experience design,” en.wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/User_experience_design. [Accessed 12 April 2026].

[18] Wikipedia, “User interface,” en.wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/User_interface. [Accessed 12 April 2026].

[19] Figma, “What is design thinking? A guide to the 5 stages and principles,” figma.com, [Online]. Available: https://www.figma.com/resource-library/what-is-design-thinking/. [Accessed 5 May 2026].

[20] Designlab, “What is the UX Design Process?,” designlab.com, [Online]. Available: https://designlab.com/blog/what-is-the-ux-design-process. [Accessed 13 April 2026].

[21] Nielsen Norman Group, “Why You Only Need to Test with 5 Users,” nngroup.com, [Online]. Available: https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/. [Accessed 13 April 2026].

[22] Tööelu OÜ, “Begin,” begin.ee, [Online]. Available: https://begin.ee. [Accessed 23 April 2026].

[23] Deputy Inc., “Deputy,” deputy.com, [Online]. Available: https://www.deputy.com. [Accessed 23 April 2026].

[24] Wemply OÜ, “Wemply,” wemply.com, [Online]. Available: https://wemply.com. [Accessed 23 April 2026].

[25] RotaCloud Ltd., “RotaCloud,” rotacloud.com, [Online]. Available: https://rotacloud.com. [Accessed 23 April 2026].

[26] Jama Software, “The Fundamentals of Business Requirements,” jamasoftware.com, [Online]. Available: https://www.jamasoftware.com/requirements-management-guide/writing-requirements/the-fundamentals-of-business-requirements-examples-of-business-requirements-and-the-importance-of-excellence/. [Accessed 25 April 2026].

[27] AltexSoft, “Business Requirements Document,” altexsoft.com, [Online]. Available: https://www.altexsoft.com/blog/business-requirements-document/. [Accessed 25 April 2026].

[28] AltexSoft, “Functional Requirements,” altexsoft.com, [Online]. Available: https://www.altexsoft.com/blog/functional-requirements/. [Accessed 25 April 2026].

[29] Requiment, “Functional Requirements vs Business Requirements Explained,” requiment.com, [Online]. Available: https://www.requiment.com/functional-requirements-vs-business-requirements-explained/. [Accessed 25 April 2026].

[30] Eclipse Foundation, “Supporting Requirements (OpenUP),” utm.mx, [Online]. Available: https://www.utm.mx/~caff/doc/OpenUPWeb/openup/guidances/concepts/supporting_requirements_B2C4D610.html. [Accessed 25 April 2026].

[31] Wikipedia, “Non-functional requirement,” en.wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/Non-functional_requirement. [Accessed 25 April 2026].

[32] AltexSoft, “Non-Functional Requirements,” altexsoft.com, [Online]. Available: https://www.altexsoft.com/blog/non-functional-requirements/. [Accessed 25 April 2026].

[33] J. Vuuk and S. Kuusik, “Backend architecture and system development of Buka: a workforce management solution for service industry companies,” Tallinna Tehnikaülikool, Tallinn, 2026.

[34] Wikipedia, “Persona (user experience),” en.wikipedia.org, [Online]. Available: https://en.wikipedia.org/wiki/Persona_(user_experience). [Accessed 30 April 2026].

[35] Figma, “Low-fidelity prototyping,” figma.com, [Online]. Available: https://www.figma.com/resource-library/low-fidelity-prototyping/. [Accessed 30 April 2026].

[36] Figma, “High-fidelity prototyping,” figma.com, [Online]. Available: https://www.figma.com/resource-library/high-fidelity-prototyping/. [Accessed 30 April 2026].

[37] Merit Tarkvara OÜ, “Merit Aktiva,” aktiva.merit.ee, [Online]. Available: https://aktiva.merit.ee. [Accessed 7 May 2026].

[38] Merit Tarkvara OÜ, “Merit Palk,” palk.merit.ee, [Online]. Available: https://palk.merit.ee. [Accessed 7 May 2026].

[39] SimplBooks OÜ, “SimplBooks,” simplbooks.ee, [Online]. Available: https://www.simplbooks.ee/en/. [Accessed 7 May 2026].

[40] Riigi Teataja, “Töölepingu seadus,” riigiteataja.ee, 1994. [Online]. Available: https://www.riigiteataja.ee/akt/12869872. [Accessed 5 May 2026].

[41] Palgakalkulaator, “Palgakalkulaator,” palgakalkulaator.com, 2026. [Online]. Available: https://palgakalkulaator.com. [Accessed 5 May 2026].

# Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

Authors, Demi Heinsoo and Kreete Keres

1 grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “UX/UI design and development of Buka: a workforce management solution for service industry companies” , supervised by Karl-Erik Karu, Viljam Puusep.

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 non-exclusive 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.

15.05.2026

[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.

# Appendix 2 – Demi Heinsoo self-analysis and personal contribution

**Self-analysis**

One of the things I want to highlight from this project is how well we managed our timeline. We planned everything in advance, set clear deadlines, and started prototyping already in September 2025, several months before the official thesis process began. This proactive approach gave us much more breathing room and helped us avoid the typical last-minute pressure of larger projects.

The biggest challenge for me was scoping the vision. We had so many ideas and possible directions that it was sometimes difficult to decide which features were essential and which would only add complexity. This required constant re-evaluation: "Is this something users actually need, or just something we find exciting to build?" Another challenge was managing thesis work alongside a full-time job, which required strong discipline, but the early start gave me the flexibility to handle this without compromising the quality of the project.

The easiest part for me was the early ideation and prototyping stage. Since we had already conducted competitor analysis and both had real-life experience with scheduling systems, we shared a clear product vision from the start, which made the early stages efficient and fast.

The most educational part of the project was learning how to think and act like a product builder. This meant constantly asking critical questions: Why are we building this? Who is it for? Does the user actually need this? Throughout the project, I followed core principles used in modern product-driven companies - taking ownership, validating ideas with real users, prioritising impact over effort, and shipping iterative improvements rather than waiting for perfection. There is no point in creating something cool if there is no real market need for it, and this mindset shaped many of our decisions.

In retrospect, one area I would improve is scope discipline. Because we had so many ideas after user interviews, it was tempting to keep adding features rather than ruthlessly prioritising. Setting firmer boundaries earlier would have made the development phase even more efficient.

Our collaboration with Kreete was one of the strongest parts of the entire project. We communicated openly, ideated freely, and were always there for each other, which prevented misunderstandings and conflicts. The earlier User Interface and Application Design course gave me a solid foundation, but the real growth happened through practical product-building - making decisions, validating them with users, and iterating based on real feedback.

**Personal contribution**

My main responsibilities included participating in UX/UI design decisions, conducting UX interviews, creating and refining prototypes, validating ideas with real users, contributing to product strategy and feature prioritization, and participating in both front- and backend discussions. During the final stage, I focused more on finalising the application itself, while Kreete contributed more heavily to finalising the written thesis.

# Appendix 3 – Kreete Keres self-analysis and personal contribution

**Self-analysis**

For me, one of the biggest challenges during the thesis project was managing the large number of ideas and possible functionalities that emerged throughout the development process. Especially after conducting UX interviews, we gained a much deeper understanding of what people in the service industry actually need from workforce management software.

Another challenge was completing the thesis while studying abroad during the semester. Working from another country and adapting to a different time zone initially created uncertainty about how smoothly the collaboration and development process would work. However, the experience turned out to be much more manageable than expected due to continuous communication, regular calls, and strong collaboration with Demi. This experience taught me the importance of communication, planning, and problem-solving within remote teamwork.

The easiest part of the project for me was developing the initial concepts and solutions, since we had already conducted market research and both had previous real-life experience with scheduling systems. Because of this, we had a clear vision of the product from the beginning, which made the early development stages efficient and straightforward.

I believe the most educational part of the project was the UX/UI design process and conducting user interviews. Conducting interviews with real users was especially valuable because it demonstrated how important practical feedback and real-world validation are during software development. The experience showed that successful products should be built around actual user needs rather than relying only on personal assumptions or ideas.

In retrospect, one area that could have been improved was maintaining more consistent effort throughout the entire project timeline. Although work on the thesis started relatively early, more steady progress from the beginning would have reduced the time pressure experienced near the final deadlines. This highlighted the importance of long-term planning and task prioritization in larger development projects.

Regarding preparation and previous studies, I felt that the User Interface and Application Design course and earlier project experiences provided a good foundation for understanding user-centred design and teamwork. However, the project also showed that practical product development, communication with users, and managing larger software projects require significantly more adaptability and planning than smaller academic assignments.

**Personal contribution**

My main responsibilities during the project included participating in UX/UI design decisions, conducting competitor analysis, organising and conducting UX interviews, creating and improving prototypes, testing usability flows, participating in both front- and backend discussions, and writing and finalising the thesis documentation. I also contributed to defining application functionalities, improving user experience flows, and validating ideas based on user feedback gathered during interviews and testing sessions.

During the final stage of the project, Demi focused more on polishing and finalising the application itself, while I contributed more heavily to finalising the written thesis and documentation.

# Appendix 4 – User research interview questions

**Manager interview questions**

*Background questions:*

- What is your sector and company size?
- What software do you currently use for scheduling and bookings? What works well and what is missing?
- What pricing structure would suit your business best (fixed fee, per user, base package with add-ons)?
- What is the maximum you would pay for an ideal solution?
- Which features do you consider most important - both those you currently have and those you feel are missing?

*After hi-fi prototype demo:*

- What was your first impression of the application?
- What stood out the most?
- What felt confusing or frustrating?
- How transparent and intuitive did the solution feel?
- Do you feel this solution would speed up internal processes such as scheduling?
- How important is pay transparency for your employees?
- Which feature in our application do you find most missing from your current system?

*Manager tasks:*

- Log in to the application
- Explain the payslip and provide feedback
- Find a specific employee's schedule and working hours
- Find an overview of your team's sick days and absences

**Employee interview questions**

*Background questions:*

- What is your role?
- How and when do you currently receive your work schedule? Which system is used?
- Are you aware of how your salary is calculated and what it consists of?
- What are your biggest pain points with your current system?
- What works well in your current system?

*After hi-fi prototype demo:*

- What stood out the most?
- What felt confusing or frustrating?
- What did you like the most?
- Did the application feel intuitive and easy to understand?
- Which feature in our system would most improve and speed up your workflow?

*Employee tasks:*

- Log in using a company code
- Swap a shift with a colleague
- Submit preferred days off
- Find and update personal account information

# Appendix 5 – Schedule generating algorithm logic

This appendix describes the first core component of the project - the schedule generating algorithm. A more detailed technical analysis can be found in the other thesis of this project [33].

At the start, the algorithm automatically generates a draft schedule for the upcoming month. This draft is displayed in the manager view, where the manager can review, manually adjust shifts, and make final changes before publishing. If the algorithm cannot fully cover all shifts while satisfying every constraint, the manager can fill the remaining gaps manually. This ensures flexibility and human oversight, guaranteeing that all shifts are ultimately covered. However, manual adjustments may result in some employees working more or fewer hours than their planned monthly workload.

Several hard constraints must always be satisfied. These include employee availability conditions such as days off, vacations, and sick leaves. Days off can be full or partial. Full days off are limited per month. For example, a manager may allow up to five per employee, while partial days off represent periods of unavailability during certain hours, such as being at school from 7:00 to 14:00.

Shift coverage is another critical hard constraint. The algorithm considers the company's opening hours, shift durations, and the required number of employees per shift. For example, a morning shift from 9:00 to 13:00 may require two employees, while an evening shift from 13:00 to 18:00 may require one. These requirements must always be met.

The main soft constraint is employee workload. Each employee has a target number of working hours per month, such as 80 or 140 hours. The algorithm distributes shifts to keep employees as close as possible to their target, with deviations permitted but penalising the overall schedule quality.

This combination of automatic generation, constraint-based logic, and manual oversight ensures the system produces practical and flexible schedules even in complex working environments.

# Appendix 6 – Wage calculation system logic

This appendix introduces the second core component of the project - the wage calculation system. A more detailed technical analysis of the system is provided in the second thesis of this project [33].

The wage calculation system is designed to ensure a transparent and accurate overview of employee remuneration. It is primarily intended for hourly-paid employees, meaning that all calculations are based on tracked working hours and the agreed hourly rate specified in each employee's employment contract. In addition, the system accounts for the employee's tax-free allowance, as defined in their submitted tax exemption declaration, including the applicable monthly amount. All calculations follow the principles established in the Töölepingu seadus [40] and Palgaseadus [3].

In contrast to traditional payroll systems, which typically calculate salaries at the end of a pay period, this system continuously updates earnings throughout the month. This enables employees to monitor their income in near real time, while the final and most accurate overview is generated at the end of the month, once all working hours and adjustments have been confirmed.

Gross salary is calculated by aggregating multiple components, including base pay derived from worked hours, vacation pay, sick leave compensation, overtime, and additional compensation for work performed under special conditions, such as public holidays, sick leave coverage and night shifts. Managers can further supplement wages by assigning bonuses, either as an hourly rate increase (e.g., +2 €/h) or as a fixed monthly amount (e.g., +200 €).

Following the calculation of gross salary, the system automatically determines the net salary by applying the relevant taxes and deductions. The calculation logic is inspired by existing solutions such as Palgakalkulaator [41].

The main objective of the wage calculation system is to enhance transparency for employees while simultaneously reducing the administrative workload associated with payroll processing.

# Appendix 7 – Login and joining the application

The joining functionality defines the process through which new employees are added to the system.

The company's manager can invite a new employee from the "Management" page using the "Add worker" option. The system then sends a unique invitation link to the employee's email address.

When the employee opens the link, they are directed to a multi-step registration process. As illustrated in Figure 43, the first step focuses on account setup, where the employee can either create a new account or log in using an existing account. Figure 44 demonstrates the second step of the process, where employee is required to enter personal information. In the third step, illustrated in Figure 45, the employee submits banking information required for salary payments.

As shown in Figure 46, the fourth step allows the employee to upload all required documentation, including the employment contract and tax-free minimum document. Finally, Figure 47 illustrated the review step, where the employee can verify all previously entered information before submitting the application.

After completing all required fields and uploading the necessary documents, the employee submits the application. A confirmation message is then displayed, and the user is redirected to the application's home page. The employee can log in only after the manager has reviewed and approved the join request.

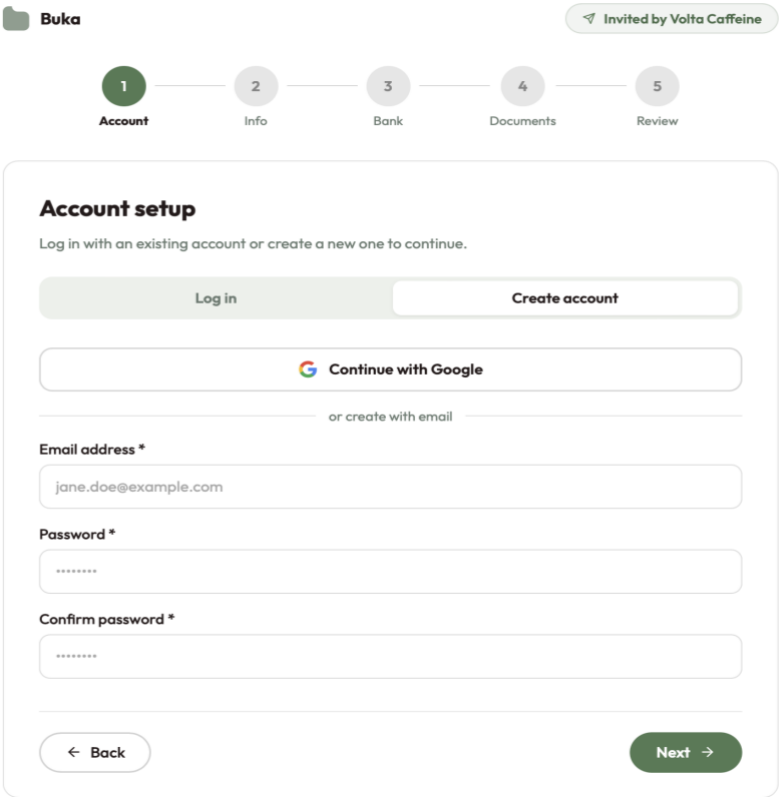


Figure 43. Employee registration process – step 1.

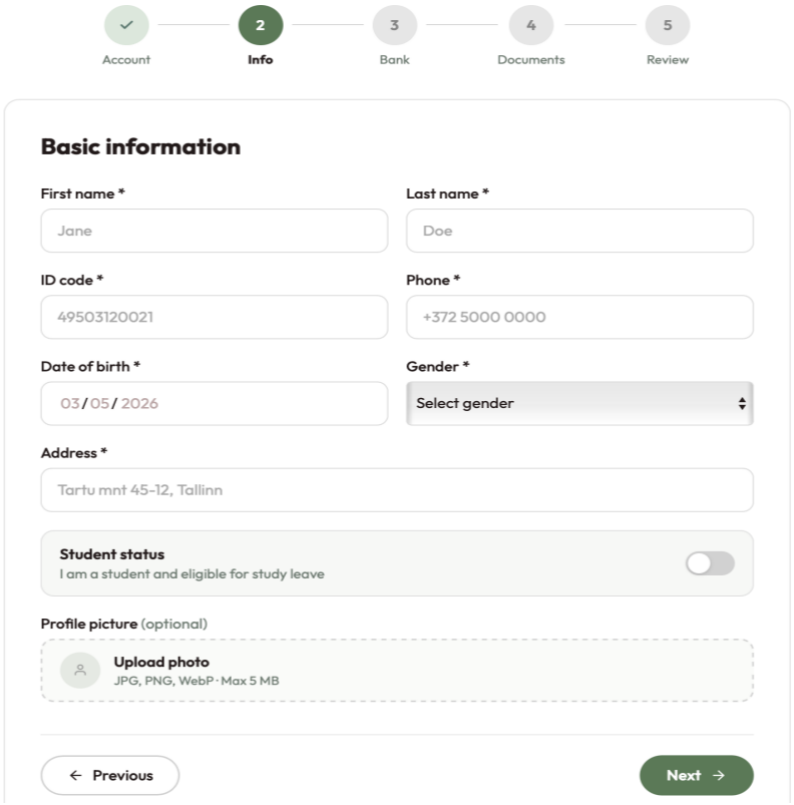


Figure 44. Employee registration process – step 2.

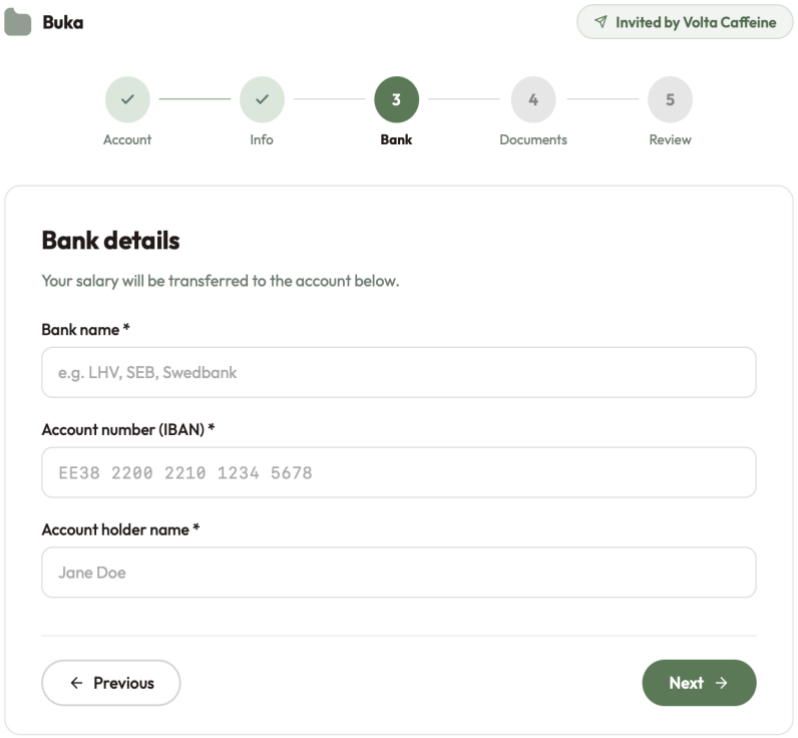


Figure 45. Employee registration process – step 3.

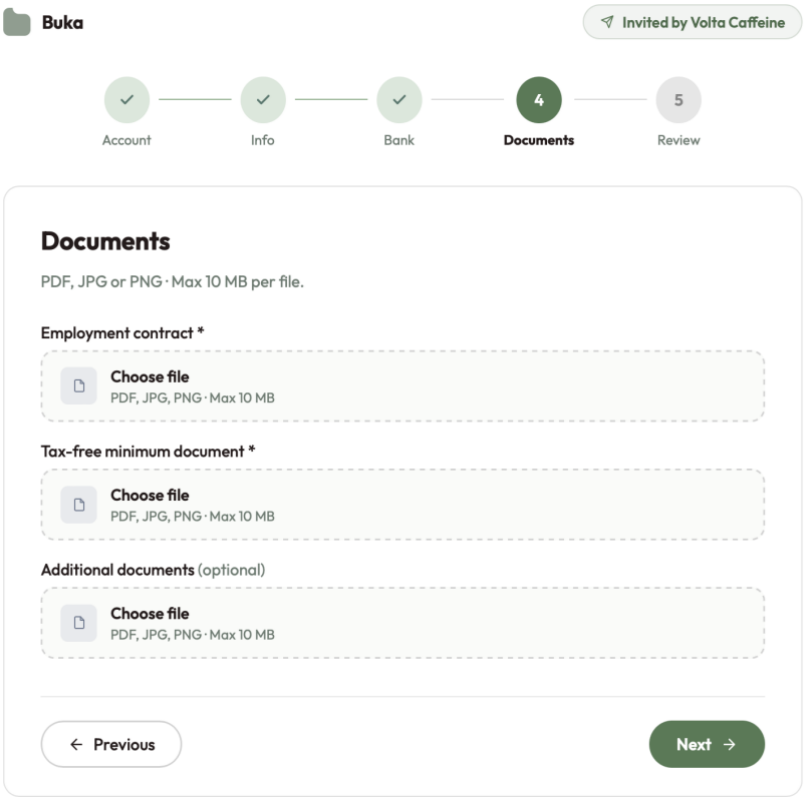


Figure 46. Employee registration process – step 4.

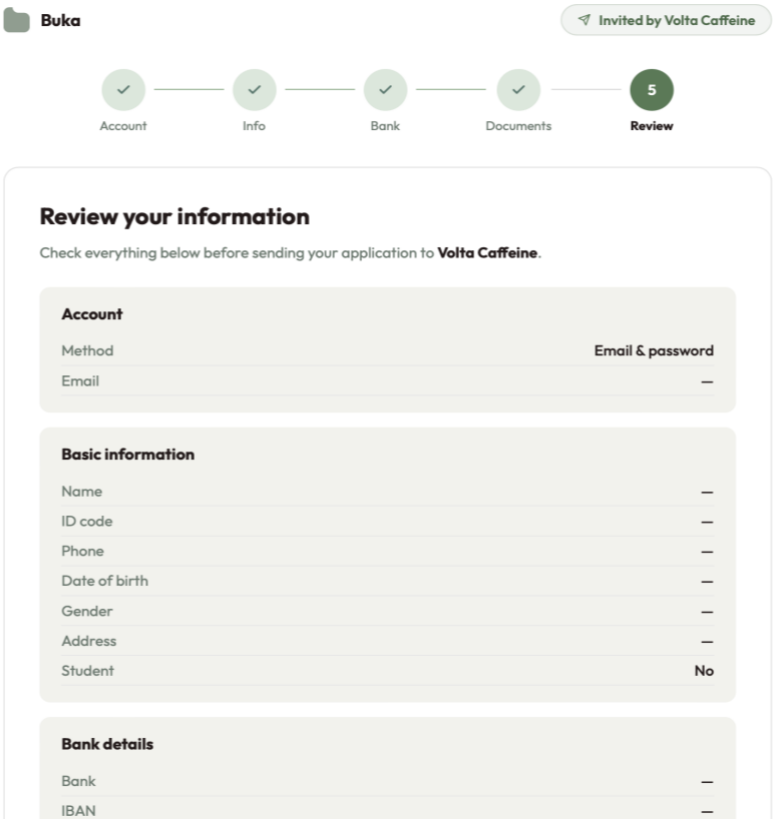


Figure 47. Employee registration process – step 5.