LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT

School of Engineering Science

Software Engineering

Ilari Sahi

DESIGNING A DECISION SUPPORT SYSTEM FOR HUMAN RESOURCE ALLOCATION IN A PROJECT-BASED ORGANISATION

Examiners: Professor Kari Smolander

Postdoctoral researcher Shola Oyedeji

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT

School of Engineering Science

Tietotekniikan koulutusohjelma

Ilari Sahi

Päätöksenteon tukijärjestelmän suunnittelu ihmisresurssien allokointiin projektipohjaisessa organisaatiossa

Diplomityö 2020

75 sivua, 4 taulukkoa, 18 kuvaa, 2 liitettä

Työn tarkastajat:	Professori Kari Smolander
	Tutkijatohtori Shola Oyedeji
Hakusanat:	ihmisresurssien hallinta, projektipohjainen organisaatio, ihmisresurssien allokointi, päätöksenteon tukijärjestelmä
Keywords:	human resource management, project-based organisation, human resource allocation, staffing, decision support system

Ihmisresurssien allokointi on haasteellista organisaatioille useilla toimialoilla. Koska työntekijät hallitsevat monia taitoja ja heillä on henkilökohtaisia mieltymyksiä, johtajien on haastavaa hallita kaikkea oleellista tietoa manuaalisesti. Tämä diplomityö tutkii ihmisresurssien allokoinnin ongelmaa projektipohjaisen organisaation kontekstissa. Yksi päätavoitteista oli identifioida tekijöitä, jotka vaikuttavat allokointipäätökseen. Toinen tavoite oli suunnitella päätöksenteon tukijärjestelmä case-organisaatiolle, jota voitaisiin hyödyntää allokointiprosessissa. Vaikuttavat tekijät identifioitiin kirjallisuudesta ja haastattelusta case-organisaation kanssa. Tekijät priorisoitiin case-organisaatiolle ja niitä käytettiin päätöksenteon tukijärjestelmän pohjana. Tutkimuksen tärkeimmät tulokset ovat avainsanoihin perustuva työntekijöiden haku ja oleellisimpien käyttöliittymäkomponenttien tunnistaminen. Päätöksenteon tukijärjestelmä tulisi integroida organisaation muiden järjestelmien kanssa, jotta se olisi kaikkein tehokkain.

ABSTRACT

Lappeenranta-Lahti University of Technology LUT

School of Engineering Science

Software Engineering

Ilari Sahi

Designing a decision support system for human resource allocation in a project-based organisation

Master's thesis 2020

75 pages, 4 tables, 18 figures, 2 appendices

Examiners: Professor Kari Smolander

Postdoctoral researcher Shola Oyedeji

Keywords: human resource management, project-based organisation, human resource allocation, staffing, decision support system

Human resource allocation is challenging organisations in various industries. Because employees today possess multiple skills and have personal preferences, it is difficult for managers to handle all the relevant information manually. This thesis studies the human resource allocation problem in the context of a project-based organisation. One of the main goals was to identify the factors that influence the allocation decision. Another goal was to design a decision support system for a case organisation, that could be utilised in the allocation process. The factors were identified from the literature and an interview with the case organisation. The factors were prioritised for the case organisation and used as a basis for the decision support system. Design science research was utilised to design the system. The main contributions of the research are the keyword-based employee search and identification of key user interface components. The decision support system should be integrated with other systems in the organisation to achieve maximum effectiveness.

FOREWORD

Finishing my journey of five years at LUT University during a global pandemic is certainly exciting. I think this year we have all learnt that you never know what the future holds. Luckily, my time at LUT has prepared me to be able to react to changes in the dynamic environment that is life. I am grateful for all the opportunities and lessons learnt over the years, and the lasting connections I have made here in Finland and abroad. I would like to thank my family and friends for the support, which made finishing this thesis possible.

15th of August 2020
Ilari Sahi

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

API	Application programming interface
DSR	Design science research
DSS	Decision support system
ERP	Enterprise resource planning
IDSS	Intelligent decision support system
PBO	Project-based organisation
REST	Representational state transfer

1 INTRODUCTION

This is a master's thesis in software engineering, conducted in Lappeenranta-Lahti University of Technology LUT. This thesis studies human resource allocation in the context of a project-based organisation. It aims to design a decision support system that managers could utilise to improve the allocation process. This chapter introduces the context and the structure of the thesis.

1.1 Background

Project management is a strategic competence for organisations, which allows them to compete effectively, reach business goals, and respond to the changing environment [1]. One of the more critical areas of project management is human resource management [2]. Human resources cannot be managed as equivalent components anymore, as each employee has a unique skill set and experience accumulated over the years [3]. Especially in software engineering, skill requirements differ greatly depending on the project position. For example, front-end developers deal with entirely different technologies compared to back-end developers. Still, managers need to find suitable employees from the human resource pool of the organisation to assemble the project team.

The human resource allocation problem is universal, faced by organisations representing different industries all over the world. Resolving conflicting resource demands is an essential responsibility of the management, making the organisations effective and sustainable [4]. The problem is even more complex for a project-based organisation (PBO), which needs to take into account various factors influencing the allocation decisions, on different levels of the organisation. Because a successful project is one that has a definitive ending [1], a PBO has a constant inflow and outflow of projects. The organisation has to make allocation decisions all the time, which can be challenging and time-consuming.

There have been many studies concerning human resource allocation, especially in the software engineering industry. However, most of the research has been focusing on allocation algorithms and optimisation of the search results [5]. Not many comprehensive system designs have been proposed, and the focus on user experience has usually been lacking. Current commercial human resource management solutions offer very simple allocation decision support, mainly in the form of employee search by skills. Moreover, a majority of managers are dissatisfied with the reliability and quality of information current information systems provide [6]. Therefore, there is room to improve the human resource allocation research by proposing a system design that offers versatile allocation decision support while providing a good user experience.

The case organisation has been rapidly growing in recent years, and managers are starting to be challenged by the amount of information required to make effective human resource allocation decisions. The current information systems the organisation has in use do not provide sufficient decision making support, and therefore most of the work has to be done manually. One of the main challenges for the organisation is that the relevant information is fragmented across multiple systems, and tools to help match employees and projects are lacking. The case organisation is a good example of an organisation that needs support in the allocation process, which has previously been mostly manual.

1.2 Goals and delimitations

The main goal of this thesis is to design and develop a decision support system that could be used to aid managers to make better human resource allocation decisions. The system is designed to aid a particular case organisation but kept at a general level so that the results could be applied to various organisation types. To design the system, factors influencing the allocation decisions need to be identified and prioritised based on the needs of the case organisation. One of the goals is also to examine what kind of commercial solutions are available to help managers with human resource allocation.

This thesis has three main research questions, which are defined as follows:

1. Which factors influence the human resource allocation decisions in a project-based organisation?
2. Are there good commercial solutions available to solve the human resource allocation problem?
3. What kind of design is required from a decision support system to help managers in allocating human resources to projects?
    3.1. How should employees be matched with projects?
    3.2. What are the user interface elements required by this kind of system?

The third research question is divided into two subquestions. The goal is to examine if there is a better way to match employees with projects than current best practice suggests. Also, user interface elements are designed in more detail to propose a comprehensive user experience for a human resource allocation decision support system. The research is based on a literature review on the subject. The decision support system is designed in collaboration with the case organisation and utilises insights gained during the literature review.

Due to the ongoing global COVID-19 pandemic, the research has to be conducted remotely. Therefore, interviews and other research activities cannot be held in person, which may slightly affect the results. The remote work tool support, however, has rapidly been improving in recent years, so no significant barriers are expected to hinder the thesis overall.

1.3 Structure of the thesis

This thesis consists of nine chapters. The introduction, which provides context to the rest of the thesis, is followed by a literature review. In the literature review, a theoretical background is provided to the human resource allocation problem following a top-down approach. Past research from project management theory to decision support systems is studied. The third chapter introduces design science research and the research methods used in the thesis. The rest of the thesis follows the design science research process introduced in chapter 3.1. In chapter four, the case organisation is introduced, and the problem is further defined. Chapter five discusses the objectives of the design science research, while chapter six describes the design and development process. In chapter seven, the resulting artefact is evaluated, and chapter eight discusses the research results. Finally, the thesis is summarised in chapter nine.

2 LITERATURE REVIEW

A literature review was conducted to gain a more comprehensive understanding of the research problem and its context. The literature review starts with an overview of project management, as human resource allocation is directly linked to the field. Second, project-based organisations are introduced to provide some context for the case organisation. Human resource management and human resource allocation are then reviewed in the context of project-based organisations. Finally, decision support systems for the human resource allocation problem are examined. The findings of the literature review are utilised in the study, particularly in the requirements specification.

Multiple sources were used in the literature review, mainly LUT University's e-resource service Finna, Elsevier's ScienceDirect, IEEE Xplore and Google Scholar. The following search terms and their combinations were used to filter the results: software engineering, software development, project management, multi-project, project-oriented, project-based, staffing, human resource allocation, decision support systems, and design science research. In addition, multiple articles were discovered from the references of other articles.

2.1 Project management

Software engineering projects are usually initiated to create new products or to update or maintain existing ones [7]. The goal of project management is to execute projects effectively and efficiently by planning, organising, measuring, monitoring and controlling organisational resources [8]. Project management also ensures that the project benefits the different stakeholders [9] and provides business value for the organisation [8]. Project managers utilise different tools, skills, techniques and knowledge in project activities so that the defined requirements are met [1]. Today there are many tools and processes available from which the project manager can select the ones that fit the project at hand the best [8]. Project management is highly contextual, and the selected methods should align with the organisational environment [10]. Effective project management tools used in one project may be ineffective or even decrease performance in another project or organisation.

The frequently cited "Project Management Body of Knowledge" defines ten different areas of project management: project integration, scope, schedule, cost, quality, resource, communications, risk, procurement and stakeholder management [1]. For software project management success, effective management of the project resources is vital. Software engineering is human- and knowledge-intensive, meaning that humans and the knowledge they possess are the most important resources in any software project [11]. Advanced tools and techniques are not of much use if the quality of the people and the quality of their management is subpar [11], [12]. Human resource management is further discussed in chapter 2.3.

The traditional view of project management has often been illustrated as a triple-constraint triangle, where organisational resources are managed in relation to project schedule, budget and scope. Prioritisation of one of the three constraints always comes at the cost of the other two. Especially in software projects, the scope of the project changes constantly as the team adjusts project requirements, affecting project schedule and budget [7]. For example, if more features are required from the finished product than initially planned, the original project deadline would likely not be achievable. Furthermore, software engineering and other high-technology projects are characterised by the low accuracy of estimations [13]. The resources should also be managed within good stakeholder relations [8], as described in figure 1.

Figure 1. Triple-constraint model of project management. Adapted from [8].

More recently, project management has shifted towards multi-constraint thinking, where prioritisation of multiple constraints change during the project lifetime based on the needs of different stakeholders [8], [14]. Software projects can have a variety of competing constraints, including security, reliability, scalability, performance, ease of use, availability and accessibility [7]. For example, a software project can be delivered on time and on budget while fulfilling the required scope, and still be unintuitive according to the end-users. It can be argued that all these different characteristics of software can be linked to the scope constraint in the traditional project management model, but it is still important to understand that scope itself is dependent on multiple constraints.

Project governance is a supporting process which aims to link projects to the organisational strategy and provide an environment in which it is possible to successfully carry out projects [15], [16]. Project governance enables collaboration and reflection [15] while applying guidelines and restrictions, to make delivery of projects predictable [16], [17]. Governance is required, as projects are rarely independent. Instead, projects are interdependent, resulting in conflicts of processes, resources, goals and objectives [16], [17]. Interdependence creates uncertainty and unpredictability, which are often countered with increased governance [4]. In software projects, where the outputs are intangible and difficult to comprehend, governance is typically emphasised to bring visibility for the stakeholders [7].

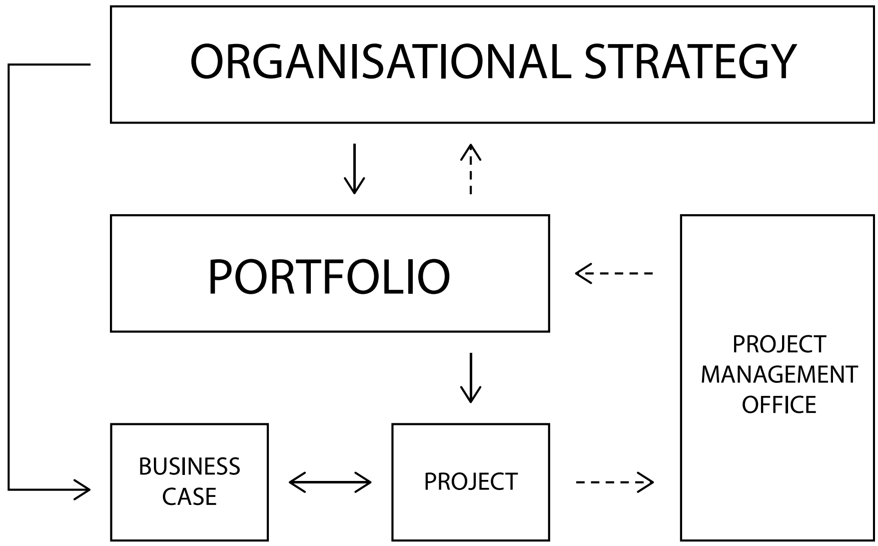


Figure 2. The relationship between strategy and projects. Adapted from [16].

Project governance can introduce different elements to the organisation, such as portfolio management and project management office [7], as pictured in figure 2. Each project originates from a real business case that is selected based on the strategic goals of the organisation. Projects are placed in portfolios, which are managed according to the strategic goals [16]. Project management office is an organisational unit that enables the extraction of data from projects so that they can be evaluated against the strategic goals of the organisation [1], [16]. Additionally, responsibilities of a project management office include standardisation of project management across the organisation, project management knowledge transfer, benchmarking and portfolio management support [1], [8].

A portfolio is a group of projects or programs that are managed together to achieve strategic goals of the organisation [1]. The goal of portfolio management is to manage the interdependencies between different projects in the portfolio, mainly balancing the competing resource demands [1], [9], [16]. Portfolio management functions include prioritisation of projects, resource allocation and identifying possible resource shortages [16]. Effective decision-making is achieved by gathering information on resource availability and developing methods and criteria for decision-making [17]. Resource allocation is further discussed in chapter 2.4.

There are three different organisational structure types – functional, matrix and projectised structure – all of which view projects differently. Functional structure consists of individual departments for each function in the organization. For example, engineers, designers and salespersons all have their respective functional departments. The functional structure is rarely suitable for project work, as projects need to flow through functional departments and collaboration between departments is difficult [8]. On the contrary, the projectised structure is organised around projects. Each project consists of full-time project members and a project manager who is responsible for them and the project outcome [1], [18].

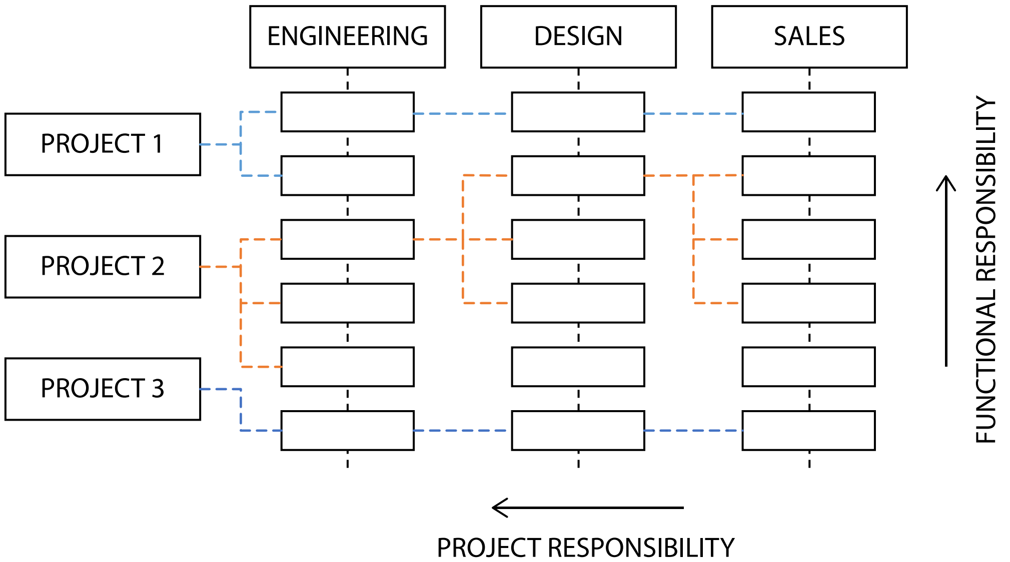


Figure 3. Matrix organisation structure. Adapted from [8].

The matrix organisation structure shares responsibilities between functional and project managers [13], as illustrated in figure 3. The project manager is responsible for the project success, while the functional manager is responsible for the technical competency of the employees in their department [8]. There are three types of matrix structures – weak, balanced and strong – weak matrix giving less authority for the project manager, and strong matrix giving almost full responsibility for the project manager [18]. In a strong matrix, projects are considered profit centres and functional departments cost centres [13]. The matrix organisational structure is commonly used in research and development organisations [19], such as software development companies.

2.2 Project-based organisations

Organisations that mainly conduct their work as projects have been defined, for example, as project-based [18], project-oriented [20] and multi-project organisations [3]. There are some differences between the definitions; for instance, project-based organisations carry out projects because they have to, while project-oriented organisations choose to organise their work as projects [20]. In this thesis, the project-based organisation (PBO) is used as a high-level term for these kinds of organisations.

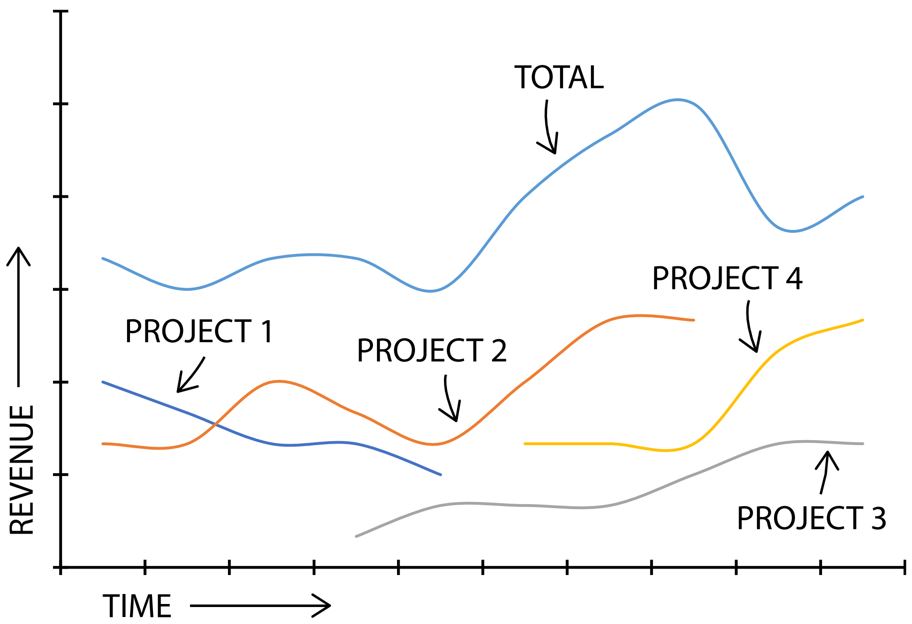


Figure 4. Revenue flow of a project-based organisation. Adapted from [8].

Typically PBOs have a shared project management culture and a portfolio or multiple portfolios containing different types of projects, while also providing supporting functions, such as accounting, through permanent units [21]. PBOs are quite dynamic compared to traditional organisations [22], [23]. Because a successful project is one that has an explicit ending [24], parts of a PBO are continuously disappearing while new projects emerge to take their place. Fluctuation in the portfolio can be directly observed in the total revenue of the organisation, as illustrated in figure 4. As income is extracted from different projects, one project ending will result in a drop in the total revenue, unless the organisation has a steady stream of incoming projects. Employees are reassigned continuously to different projects, and every time the organisational structure changes slightly. PBOs can follow various organisational structures [18], [25], but they usually organise themselves as matrix organisations [3], [26]. The projectised structure has been perceived as more effective than the matrix structure [26], but PBOs benefit from a matrix structure in situations where not all project members are assigned as full-time members. The matrix structure enables employees to switch from project to project with additional support from the functional department.

There are generally four levels of management in a PBO: single project management, portfolio management, functional management and management of the entire organisation [27], [28]. Portfolio managers, together with functional managers, ensure that the resources of the organisation are used efficiently and in the right projects. Everyone in the organisation must comply with the fact that the organisation is project-based [23]. The organisational strategy, policies and practices should all be shaped and developed to support project work [29]. Top management needs to make sure that the organisational context is suitable and supportive, which helps in continuous and successful execution of projects. Useful PBO management tools and practices include organisation level resource planning [27], portfolio steering committee [21], project management office [30], formalised knowledge transfer process [20], [31], flat hierarchy and customer orientation [21]. Typical reaction from top management to gain more control over a PBO is to introduce more hierarchy, but it often is more harmful than useful [25]. The dynamic nature of the organisation benefits from flat hierarchy and quick decision-making processes. Management of a PBO should be supported by information and decision support systems [25], [28], [32], which are further discussed in chapter 2.5. The effectiveness of PBO management can be measured, for example, by resource productivity and organisational learning [33]. Competent governance is required to keep all the individual projects aligned with the organisational goals and objectives [15], [16].

Selection and prioritisation of projects are key activities of any organisation that uses projects as a way of working [17]. For efficient operation, organisations need to take on projects that utilise its resources optimally and bring maximum value to the portfolio [34]. The selection process should be open, consistent and fair to all stakeholders, while also considering feasibility and alignment with organisational strategy [3]. The process should also take into account the existing portfolio and the risks new projects might bring into the portfolio [35]. Portfolio planning process should be established and executed periodically to retain control over the portfolio and its projects [33]. If the portfolio content changes frequently or the portfolio projects are unstable and cause unplanned resource usage, the portfolio planning process should take place more frequently [27]. Portfolio planning requires project plans of the individual projects and information about resource availability from the functional departments [27]. Different project priorities bring additional complexity to PBO management [3]. Because prioritisation is a difficult task, some companies choose not to practice it at all [8], while others assign top priority for all projects [35].

Interdependencies are the source of complexity in PBOs, affecting organisational performance and requiring strategic portfolio management [35]. Projects can share human resources, technologies, information and even work towards the same objectives [17], [33], as illustrated in figure 5. Projects also need to integrate with management and reporting systems [3] and communicate with each other [27]. Even though two different projects have different goals, they may still use shared resources. Changes or delays in one project affect the whole portfolio and can cause negative effects to the schedules of current and future projects [10], [36]. The uncertain environment can lead to a competitive behaviour between project managers over scarce resources, who all want their own projects to succeed and be completed on time [31]. Adding new projects to the portfolio can also affect existing projects negatively, if the available resources are inadequate [37]. The lack of interdependency management makes it difficult to have a clear overview on the overall situation, resulting in a feeling of chaos [38].

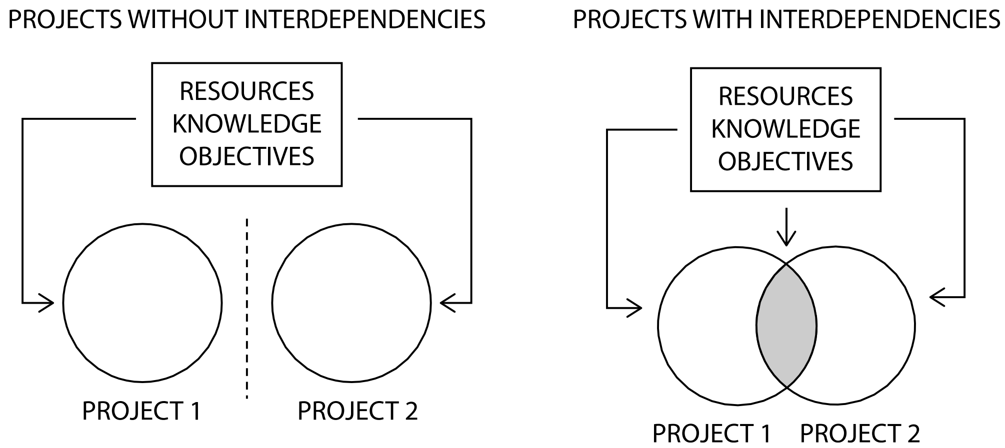


Figure 5. Two projects without and with interdependencies.

PBOs should enable sharing of project information and information about past decisions so that the knowledge gained during a project can be preserved and organisational learning can take place [22], [31]. Having an organisational knowledge base, from which different projects can draw information from, helps in avoiding past mistakes and improving technical best practices [31]. Motivating knowledge sharing in a project-based organisation is a difficult problem to overcome because objectives usually focus on individual projects and the improvement of the organisation as a whole is ignored [22].

2.3 Human resource management in project-based organisations

Typically, organisations have resources of seven different types: raw material, facilities, equipment, technology, money, information and human resources [8]. One of the popular corporate governance theories, the resource dependence theory, claims that “the key to organisational survival is the ability to acquire and maintain resources” [4]. Human resources are the most important for knowledge-intensive organisations that practice research and development, such as software engineering organisations, as the results are highly dependent on teamwork and the intellect of the personnel [1], [7], [15], [19], [39]. The cost of software is measured in person-hours or -days, further fortifying the fact that the development process relies on human factors, such as skills, abilities, leadership and organisational culture [7]. Therefore, the problems in these kinds of organisations also tend to be sociological, rather than technical [24]. Ineffective communication and human resource management are common causes of project failure [5], [39]–[41]. However, it must be noted that some studies have concluded that the personnel factor does not have a significant impact on project success [42]. Organisational studies are heavily context-dependent though [10], and different studies yield different results. It is also difficult to evaluate how personnel affect project success, as the personnel factor is ambiguous and can mean different things to different people. Even if a project succeeds because of technical excellence, in software engineering, it ultimately comes down to the skills and abilities of an individual. Developing the skills of an employee should be perceived as an investment, rather than an expenditure [24]. Each project adds to the skill set of an employee and makes them more valuable to the organisation. If that employee were to let go, the investment would be lost and potentially captured by a competitor.

Management of the human resources of a project include identification of resource needs, obtaining resources and managing them for successful project completion [1]. For project managers, soft skills, such as communication, are of the utmost importance [2]. Software project managers should ensure that the team is compatible and any communication barriers are dealt with so that the team members can focus on the technical execution [11]. One important managerial aspect to grasp is the uniqueness of an individual human resource. Managing people like physical machine components is a guaranteed path to project failure [24]. Software engineering personnel are characterised by scarcity, heterogeneity and non-substitutability [39]. Both technical skills and domain knowledge are required of software engineers, making every engineer different by abilities and experience [5]. As a result, replacing a software engineer with another can affect the project budget and schedule, as opposed to replacing a simple component on a machine [3], [22]. There are other reasons for non-substitutability as well. For example, in the consulting business, individual employees are usually contracted by the customer. If the organisation wishes to transfer employees between projects, the contracts between different customers need to be renegotiated.

Human resource management is a core process for project-based organisations [29]. The importance of human resources is emphasised by the fact that different projects require non-standardised sets of skills and a functional project team [22]. Therefore human resources are intellectual and social capital for the organisation that also enables the organisation to react to changes in the environment [28]. It can be argued that every PBO is in the communication business, as project work requires constant and effective communication between different stakeholders [24]. Not only does the actual development work require effort, but also the management of interdependencies between different tasks and projects, which is mostly communication, takes effort [7].

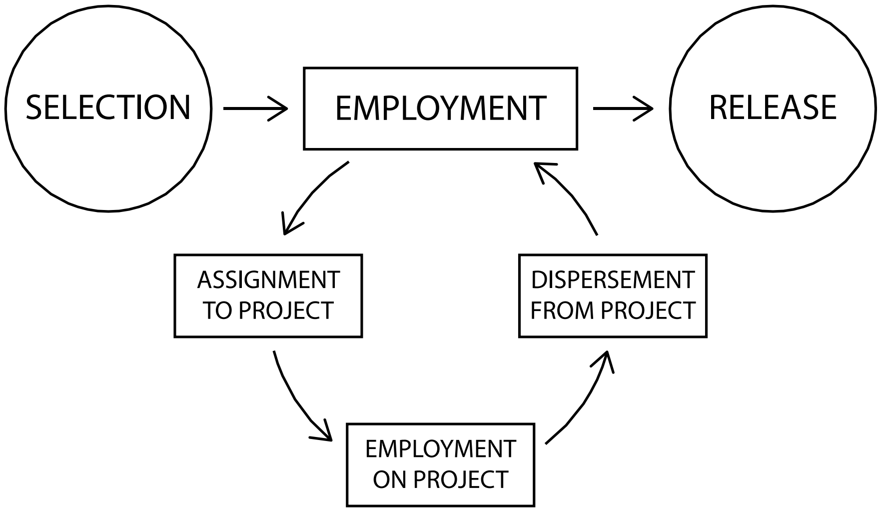


Figure 6. Human resource cycle in a project-based organisation. Adapted from [29].

If a PBO is organised as a matrix structure, human resources are owned by functional departments and assigned to projects directly or via portfolios [33]. Human resource management needs to happen in the functional department and during the project [20]. As described in figure 6, the functional department is responsible for selection and release from the organisation, whereas a project is responsible for assignment and release from the project. The project cycle is repeated continuously and even simultaneously with other project cycles during the employment in the functional department. Appraisal, reward and development need to happen in the department and on the project individually. However, appraisal in the department should always be based on the appraisal on the projects [20]. It is the temporary and dynamic nature of projects that create a unique context for human resource management in PBOs compared to other types of organisations [29].

PBOs regularly experience periods of undercapacity because management wants to maximise employee utilisation [3]. New projects are added to the portfolio without the available resource capacity, which often leads to overtime work for the current employees. It is the management's responsibility to make sure that changes in the portfolio or single project scope or schedule do not create excessive amounts of stress for the employees [22]. The problem becomes even more complex when the employees work on multiple different projects concurrently. Lack of commitment [3], uncertainty, role overload, and conflict are typical human resource management problems in PBOs [22]. Ensuring appropriate workload for a single employee requires communication and coordination between project, functional and portfolio managers. Constant undercapacity also leads to dismissal of lower priority projects that always fall behind the higher priority projects in resource allocation [3], [10], [33].

To combat fluctuating resource demand and lack of required skills, organisations can employ external consultants and freelance workers [8], [22]. Organisations can also reserve some amount of specialised human resources as a backup if there would be a sudden demand for such resources due to introduction of a high-priority project [36]. Having backup resources is not typical behaviour though, as management wants to fully utilise possessed resources, leading to undercapacity rather than overcapacity.

Employees of a PBO can work towards multiple objectives of different projects simultaneously, which requires strong commitment and communication [33]. Reaching a flow state, which is extremely important on knowledge-intensive fields such as software engineering, is difficult in a PBO due to constant interruptions [24]. Large amount of meetings, information overload, unclear responsibilities and lack of prioritisation are all problems that negatively affect the performance of employees working on simultaneous projects [38]. Switching from one project to another is not instantaneous. On the contrary, employees working on multiple projects can mostly spend their day inefficiently by constantly "switching gears" [24].

Employee wellbeing is an important, but unfortunately often overlooked aspect of human resource management in PBOs [29]. As projects start and finish, the organisational structure is in a constant state of change as resources are reconfigured. The uncertainty that employees may feel can be mitigated by creating processes for project assignment and release, while also making sure that project outcomes are linked to the long-term career of an employee [29]. When a project ends, the released employees can move straight to a new project, go through training, do internal technical or process development, or "sit on a bench" [22]. The decision should be discussed with the employee since it directly affects their wellbeing. Caution should be exercised on the management's side as well because a dissatisfied employee without anything to do is likely to look for work from somewhere else [29].

Organisations must ensure that the career goals of the employees are considered in the long-term by the functional departments, as opposed to only focusing on the aspirations of the employees in the short-term projects [9], [20], [22], [28]. The career of a PBO employee is built on successive projects, which individually are merely learning experiences, but together form a coherent path, if carefully managed [29]. As opposed to ascending in the hierarchy, career development in a PBO can focus on giving employees more responsibility and strategically more important and complex projects to work on to display advancement on their career [22]. Because PBOs benefit from a flat hierarchy, it is not practical to promote the traditional, vertical career development. Instead, expanding employee's skill set [22] and strengthening their strategical importance in the organisation leads to a more meaningful career in the dynamic context of a PBO.

2.4 Human resource allocation in project-based organisations

Project team collectively works towards a shared goal [1] and as such, human resource allocation, or staffing, lays the foundation for a project's success or failure [24]. Software engineering especially is highly dependent on successful resource allocation because of its knowledge-intensiveness [5], [19], [43], since the project tasks and allocated resources are interdependent [39]. Failed resource allocation often leads to a reduced scope, prolonged schedule, lowered customer satisfaction and even project failure, which calls for resource allocation process development [1], [44].

On the portfolio level, human resource allocation process should take place when new projects are initiated and old ones completed, and if there are any personnel changes, such as retirement or hiring of new staff [37]. Human resource allocation can also be considered if career goals and personal preferences of employees change [45], but it is not as crucial in PBOs as projects are only temporary. In the context of an individual project, human resource allocation should be executed periodically during the project lifetime [1], [5], [37]. Human resources are allocated when the project is initiated, but allocation should be revisited when project scope, schedule or budget changes because human resource needs can be affected as well.

There are three different levels of resource allocation planning: short-, medium- and long-term [19], [35]. Short-term planning happens at the operational level, medium-term at the tactical level, and long-term at the strategic level of the organisation. Whereas short-term planning focuses on individual projects and employees, medium- and long-term planning focus on the portfolio and organisational perspective [35]. Short-term planning is mostly weekly task allocation for team members inside an individual project [19]. Medium-term planning takes place on the portfolio level and may be conducted quarterly [35]. It aims to allocate the available resources to the projects in the portfolio, so that project managers are aware of their team and can communicate the resource configuration to other stakeholders in the project [19]. Medium-term planning should consider the upcoming projects so that employees are not allocated hastily, as well as the ending projects that release resources back to the resource pool. The objective of long-term planning is to determine what kind of resources different functional departments need and in what quantity [19]. Long-term planning can take place yearly and should be part of the overall business planning process [19]. To achieve organisational goals set by the top management, short- and medium-term planning need to be linked to the strategic, long-term planning [19], [35]. For example, medium-term planning needs to follow constraints set in the long-term plan, while long-term planning must address shortcomings identified in medium-term planning. If the organisation does not utilise medium- and long-term planning, the allocation will mostly be reactive to project-specific issues, which makes achieving the organisational objectives challenging [46].

The human resource allocation process begins with the estimation of resources the project requires, based on the project plan and schedule [1]. The goal is to determine what kind of resources the project needs and when. Resource requirements depend on the scope, schedule, budget and priority of the project [7]. Project scope determines the skills and types of team members the project needs. Software projects may need operational, financial and legal skills in addition to purely technical skills [47]. The project may also need various soft skills, such as negotiation, depending on the project stakeholders [47]. Lines of code [48] or person-days and -years are often used to estimate the scope quantitatively. Project schedule affects the resource allocation timeline. Some resources may not be needed at the beginning of the project, and resource demands can escalate towards the end of the project lifecycle [8]. In addition to restricting the team size, project budget can also restrict the level of experience available for the project [5]. Senior software engineers are more expensive compared to junior engineers, so low-budget projects are likely to be staffed with low-cost engineers. The priority of a project can be determined based on the organisational value of the project [37]. The project requirements and restrictions will change during its lifecycle, and the resource allocation process must be revisited accordingly [1], [7]. Especially software engineering projects often experience the emergence of new requirements as the project progresses [44].

Typical human resource allocation flow of a project is described in figure 7. In software projects, requirements specification takes place at the beginning of a project, which does not need as many resources as the actual development, which takes place towards the middle and the end of the project lifecycle. As the need for resources grows, employees are brought along from other projects that do not require their contribution anymore. When the project nears the end of its life, and most of the development work has been completed, resources are released back to the resource pool so that they can be utilised in other projects.

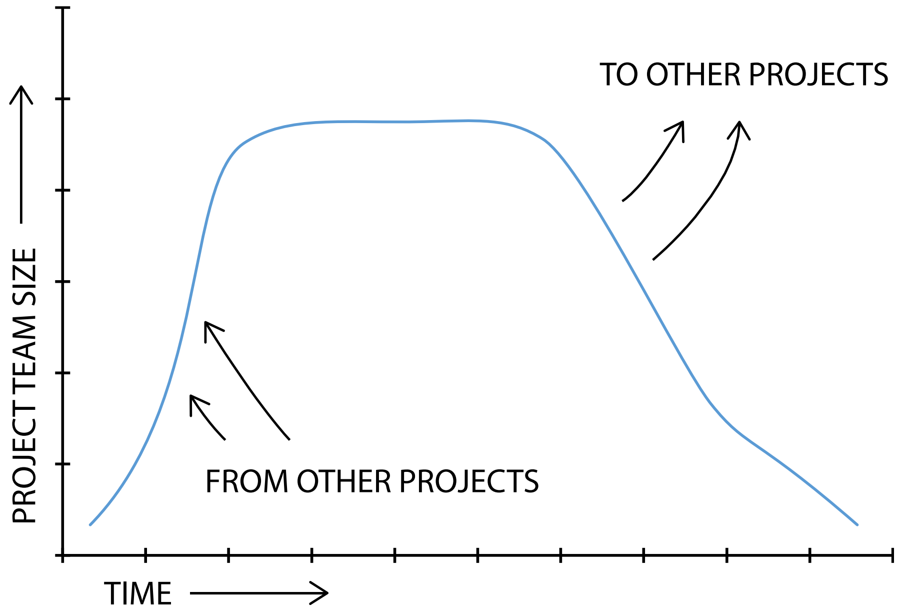


Figure 7. Human resource allocation level during the project lifecycle. Adapted from [8], [18].

Human resources are allocated to a project based on the project resource estimation and information available of the organisation's human resources, including skills, abilities and experience [1]. Selecting the right employees to the right projects is a complex issue and requires multi-criteria decision analysis [1], [34]. Often a perfect fit for a specific task is not available, so management needs to decide on the second-best option [48]. Various factors need to be taken into account when selecting suitable employees for a project, including employee skills and skill levels, possessed certificates, overall productivity and per skill, knowledge of project domain, learning and forgetting effects, working hours, salary and other expenses, level of commitment, reliability, experience in working with the project stakeholders, career objectives, personality, as well as organisational skill set development needs [1], [2], [8], [22], [37], [44], [45], [47], [49], [50]. The learning effect denotes how fast an employee can learn a particular skill, while the forgetting effect measures, how quickly a certain skill level deteriorates if that skill is not used and maintained [50]. It is often true that mastering one skill makes learning a related skill easier [48]. Project managers may also favour employees of whom they already have positive experiences from previous projects [8]. Different factors are weighted differently, depending on the organisational strategy and project needs [1]. The factors should be considered on the individual and organisational level [29]. For example, there may be a lack of a particular skill set in the organisation, and a project comes up with an opportunity to develop skills included in that set. An employee fitting the skill set should be selected and allocated to the project so that the organisation gains competence by strengthening that particular skill set.

Considering whether specific project roles need to be filled full- or part-time is essential [1]. If some task requires only a small or irregular contribution, it is likely not sensible to have a full-time team member for that position. However, one of the objectives of human resource allocation should be to reduce the project-scatter-factor. Project-scatter-factor defined as

$$project\text{-}scatter\text{-}factor = \frac{number\ of\ team\ members}{project\ duration\ in\ person\text{-}years} \tag{1}$$

describes the ratio between the number of team members and the scope of the project measured in person-years [19]. If the number of team members is high compared to the project scope, the project-scatter-factor will also be high, meaning that more employees are allocated to a single project task. The project-scatter-factor should ideally be lower than 1.5 [51], suggesting that a maximum of 1.5 employees are performing a single task. A high project-scatter-factor implies that majority of the team-members are part-time members, which poses a risk of lacking commitment and low efficiency [19]. A balance is needed between full- and part-time team members so that the continuity of the project is guaranteed, but also specific expertise can be provided as needed [7]. It should be communicated to the project stakeholders if some team members are working only part-time on the project, and the capacity rate they contribute [8].

Project-scatter-factor relates to the famous Brooks's law, which states that "adding manpower to a late software project makes it later" [12]. In general, a large project team means that there is more intercommunication, coordination and training [1], [12]. When employees are added to a project after its initiation, they need to be trained, reducing the effort spent on the project tasks [48]. Careful management is required so that the project manager can provide a non-disruptive working environment for all the team members.

Recently more focus has been put on the personality of an individual employee in the human resource allocation process [5]. A project can benefit more from a less skilled team member who fits well in the team, than from a competent employee who does not fit at all [5], [7], [34]. Personality-wise, the right employee in the right project and team can affect team cohesion, risk of conflict, productivity and employee satisfaction [5]. One option to quantify the personality factor is the introduction of personality types in the resource allocation process. However, collection and storage of personality traits can be seen as intrusive by employees and should be utilised with great care [5].

It is recommended that management alone is not responsible for the allocation process. It should also involve the employees that are allocated by making the process transparent and well-defined [49]. Employees work more effectively if the project matches their personal preferences and career objectives [34]. If a project is initiated, that is an excellent match for a specific employee, that employee should be allocated to the project, even if it leads to a release from another project [22]. The employee gets to develop their core skills, which benefits the organisation and leads to higher employee satisfaction. The career path of an employee, which is planned jointly by management and the employee, must be considered in the allocation process. If more employees are allocated to a project after the initial allocation, it is advisable also to consult the project team, so that the most effective team can be formed and maintained [7].

The human resource allocation process in project-based organisations aims to maximise the value of employees across all projects in the portfolio [35]. Interdependencies characterise PBOs, and often employees are shared among multiple projects. Problems related to resource allocation are common in PBOs [26], [41], [46]. For example, if some project fails to release its resources after the scheduled due date, it creates a domino effect and delays upcoming projects that depend on those resources [3], [46]. This complexity must be taken into account in the human resource allocation process, and re-allocation should take place when needed. The allocation process should also be dynamic and flexible so that it can react quickly to the fast-changing portfolio of a PBO [19], [35]. Long- and medium-term human resource planning are essential in PBOs so that the big picture is not lost [3]. Often problems arise from the fact that projects are scattered, and there is no centralised information available of them, or of the skills and resources the organisation has [41]. Information systems, discussed in chapter 2.5, can help with the overall coordination. Few companies employ such systems [22].

The general problem PBOs face is that its employees are allocated to too many different projects concurrently [46]. PBOs see projects as income and other operations as an expense. If an employee happened to have some free time, they are likely to be allocated to a project, rather than sent to perform internal process improvement or training [46]. Employees working in multiple projects need to constantly switch between different contexts and objectives, accumulating effort not spent on a project and resulting in loss of productivity [7], [52]. Having employees work on a single project can drastically improve their productivity [2], [7], [51], [52]. It also makes the human resource allocation and project management processes easier [19], [22]. However, some employees may work more efficiently when assigned to multiple projects [8], so ultimately the allocation decisions should be employee-specific with some level of organisational guidance.

In small- and medium-sized PBOs that follow the matrix structure, it is typical that functional managers are also project managers [8], [18]. However, these managers can influence the allocation process and allocate the best resources to the projects that they are managing, on the expense of other projects in the portfolio [8]. The process should be made as transparent as possible to mitigate the risk of misuse of the process.

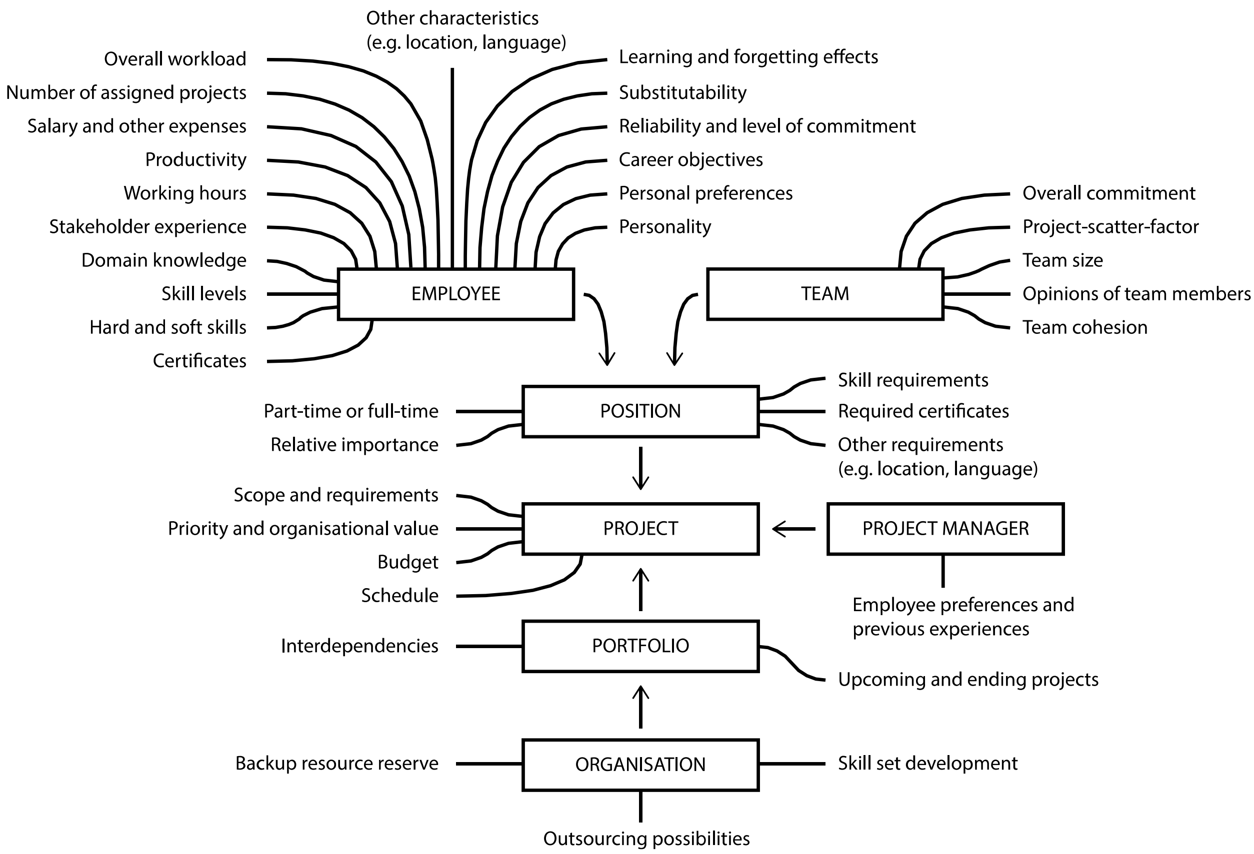


Figure 8. Factors influencing human resource allocation in a project-based organisation.

Figure 8 summarises the different factors influencing the human resource allocation process identified during this literature review. The factors are categorised by the perspective: from the individual employee to the organisational level.

If the human resource allocation process lacks relevant information and is executed hastily, there is a high probability of inappropriate decision-making [49]. Therefore, great care should be taken during the allocation process. The results of the human resource allocation process include updated project schedule, resource assignments and resource calendar [1]. From the resource calendar, stakeholders can verify what the project resource configuration looks like at specific points in the project lifecycle. If the resource configuration changes, the outputs need to be updated. After all, plans and documentation are useless if not up-to-date [19].

2.5 Decision support systems for human resource allocation

Information systems extend the cognitive capabilities of managers and have become important tools for organisations [53]. In general, information systems handle large amounts of data to help managers with management and planning activities. Whereas a basic information system focuses on collecting, maintaining and displaying data, a decision support system (DSS) applies analytical models to this data to support managers in decision-making [54]. A DSS can support managers in various phases of the decision-making process, such as gathering and analysing data, forming alternative options and selecting the most favourable one [55]. DSSs can especially help in solving semi- and unstructured problems, which opposed to structured problems, are not routine or repetitive, require human judgement, and no standard solutions exist for them [55]–[57]. As the name implies, a DSS will not implement any decisions in isolation but supports the user in the decision-making process [57]. The user will ultimately decide which decision will be implemented and on what terms.

A DSS inherently consists of three different components: data management, model management and a user interface [58]. The data management component has access to the data and knowledge related to a specific decision. A common problem in decision-making is the lack of information or access to key data, which highlights the importance of the data management component [58]. The quality of decisions is also directly related to the quality of the data [6]. Equally as important is the model management component, which handles the decision models that are applied to the data. The models are used as a basis for the decision-making, so modelling different decision-making processes is a crucial activity in the development and maintenance of a DSS [55], [58]. Also, an intuitive user interface is required so that managers and other users can utilise the DSS in practice. Figure 9 describes the basic architecture of a DSS, with the addition of intelligence.

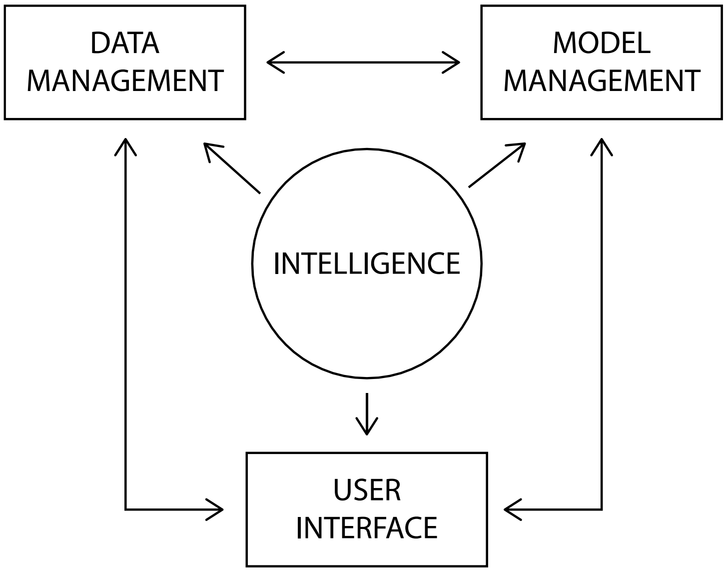


Figure 9. Architecture of an intelligent decision support system. Adapted from [55].

A DSS augmented with intelligent components, such as machine learning, is sometimes called an intelligent decision support system (IDSS) [57], [59]. The amount of information available for managers is increasing rapidly due to digitalisation, which increases the complexity of decision-making. An IDSS with data mining capabilities can significantly improve the effectiveness and efficiency of the process by sifting through massive amounts of data automatically [57]–[59]. Artificial intelligence can also be applied to modelling, for example, in the form of genetic algorithms and machine learning in general. User interface can benefit from intelligent methods as well, such as natural language processing.

Utilisation of a DSS will improve the fairness and consistency of the decision-making [57]. Whereas humans can introduce bias and forget details crucial to the decision-making [59], a DSS will include all the relevant data in the analysis. DSSs can help organisations to gain a competitive advantage [54], improve productivity and make information utilisation more efficient and effective [56]. Especially in dynamic environments, such as project-based organisations, DSSs improve the agility of the organisation and free managers to other tasks [58].

The human resource allocation task is a difficult one, and information systems should assist management in its execution [34], [60]. Software tools and systems help to make the process more efficient and effective [5]. There is evidence that DSSs perform better at the allocation process than expert judgement alone [32]. However, the value of expert judgement should not be understated. The human resource allocation process should be performed by management but supported by a DSS [45], [60]. The supporting role of DSSs should be emphasised to mitigate the risk of managers seeing them as a threat instead of a tool [61]. Benefits of a human resource allocation DSS include increased employee involvement, reduced human error, and reduced effort required for the allocation process [45], [49], [61]. To increase employee involvement employees themselves could input data about their career objectives and personal preferences into the system, which can then be used in the allocation decision. As a result, possible human error is reduced, since the DSS takes into account all the different characteristics of an individual employee, which might be forgotten or neglected by a manager.

Different factors affecting the allocation process need to be measured to utilise decision support systems. However, few organisations have processes or frameworks to perform such measurement [32], [43], [45]. For example, estimating skill levels, level of commitment, personality, and team cohesion is challenging without an organisation-wide measurement framework and guidelines. Productivity rate especially is a difficult factor to evaluate, thus in past research it is often considered equal for all employees [43]. In more complex setups, productivity rate can be set individually for each employee, calculated based on experience, or be dependent on individual activities [43]. Additionally, managers find it difficult to quantify the kind of employees and skills that are required for a project [45], [61]. Some data can be retrieved from other information systems in the organisation, such as payroll systems or enterprise resource planning systems [62]. However, most likely, the data is not sufficient in the beginning. It should be investigated whether all the necessary criteria can be aggregated from the existing data, or if there is a need for collecting entirely new data. Ethical aspects of a human resource allocation DSS should be considered as well. If the DSS can allocate employees to projects based on highly personal factors, such as personality, the privacy and confidentiality of the system must be guaranteed [45].

In a dynamic environment, keeping up with the skills and abilities the organisation personnel possess is a significant challenge [49]. This is especially true in software engineering, as employees often have a multitude of skills that do not necessarily relate to each other [50]. Software engineering is evolving rapidly, and the best-practice technologies and tools may change every year. This forces software engineers to continuously learn and develop new skills, adding to the catalogue of skills they already have. It is difficult for managers to evaluate the level of experience engineers have on specific skills [5], so individuals should evaluate their own skill levels.

A database of all the different skills and their relation to each other could also be maintained. Experience in some skills may make learning other skills easier, which could be useful, considering human resource allocation [48]. If a project requires a specific skill, but no employees that possess it are available, employees with similar skills could instead be allocated. Other aspects associated with skills are the learning and forgetting effects. An employee has to spend effort on recalling a skill if it is not used for some time, even though they would have a high level of experience in that particular skill [50]. Employees also learn new skills at a different pace, depending on their existing skill set and other individual factors. The learning and forgetting effects are employee-specific and challenging to measure but could provide more optimal results if included in the allocation process.

Table 1. Methods applied to the human resource allocation problem [5].

Field of study	Method	Description
Mathematical modelling	Linear programming	In linear programming, a linear function is minimised or maximised to find an optimal solution.
	Probabilistic modelling	Probabilistic modelling focuses on predicting the future by calculating the probabilities of certain situations based on historical data.
	Queuing theory	Queuing theory is usually used to minimise project delays by simulating a resource demand queue.
	Combinatorial optimisation	In combinatorial optimisation, multiple elements are evaluated to find an optimal solution. Constraint satisfaction is an area of combinatorial optimisation.
Computational intelligence	Evolutionary algorithms	Based on natural evolution, evolutionary algorithms evaluate each possible solution, passing the best ones to subsequent evaluation stages.
	Swarm intelligence algorithms	Swarm intelligence algorithms examine how individuals in a swarm, for example, projects in a portfolio, interact with each other to reach a common goal.
	Fuzzy logic	Fuzzy logic methods imitate human decision-making process by choosing an optimal solution based on vague and ambiguous information.

Modern human resource allocation processes seek to optimise the allocation results to maximise the value extraction from the organisation's resources [1]. The human resource allocation problem is a constraint satisfaction problem [43], classified as operations research [5]. It is an NP-hard problem [50], [60], meaning that it is extremely difficult or impossible to find the best solution in a reasonable amount of time. Different methods in mathematical modelling and computational intelligence have been used to tackle the problem, introduced in table 1. Many of the methods are search-based, which iteratively approach the optimal solution, usually resulting in a near-optimal solution. [60], [63].

The search for optimal candidates should begin with a rough selection based on necessary criteria, which is then filtered further and ranked based on the secondary criteria [45], [61]. For example, it might be vital for a position to have experience in backend development, while frontend development experience is not necessary, but a bonus. Sometimes a position may require a skill set that is not present in the organisation. In general, it is better to search for candidates that would fit the position the best, not candidates that possess the exact skill set. A degree-of-fit score could be given for each employee based on the project criteria, which is then used as a basis for the allocation decision [61]. The decision support system should also allow the use of different optimisation approaches depending on the project [50]. In some projects, it makes sense to minimise cost, whereas some projects need to be completed as quickly as possible [5]. In some cases, project managers want to minimise both the cost and duration [39]. Teams can be formed with the objective of creating the smallest team, most qualified team, the cheapest team, or the fastest team [43].

Human resource allocation optimisation tools are not widely adopted in the industry, due to lack of practical evidence, and insufficient focus on graphical user interface and integration with existing systems [32]. Especially project-based organisations lack such tools, because tool support for the allocation process on the organisational level is still low [22], [26], [64]. Several limitations also remain in past research. For example, employee changes are not taken into account [37], and the fact that employees can be assigned to multiple projects is dismissed [44]. Preferences of the employees are rarely considered, and social factors, such as personalities and team cohesion, remain out of scope. Though, as we can see from figure 8, there are numerous different factors affecting allocation decisions, which makes it extremely difficult to develop a perfect system. Considering all of them makes the optimisation problem complicated. The organisation should decide which factors it deems the most important and focus the data collection and optimisation effort on those factors specifically.

There are many commercial project and human resource management products and services that include some level of help for the human resource allocation problem. To name a few, Microsoft Project [65], Silverbucket [66], Hub Planner [67], Mavenlink [68], 10,000ft [69], Zoho People [70], Teamdeck [71], Saviom [72], and Clarizen One [73] are examples of such services. All of them are very similar in functionality, offering a very basic level of decision support for the allocation decision. Project-based organisations are generally considered well, as resources can be allocated simultaneously to multiple projects. Most of the services offer visual guidelines of the total workload of resources, helping managers to identify employees that are allocated to too many projects and have a heightened risk of a burnout. Employees can be filtered based on various criteria, such as location, availability, and skills.

However, there are several limitations in the existing commercial services. First of all, employees need to be searched individually for each position. If a project needs, for example, a designer, a backend developer and a database engineer, the services do not enable the search of all the positions at the same time. In general, the positions are considered individually, which prevents the result optimisation by cost, duration or team cohesion. The employee filtering criteria are overall very limited. Most services allow the management of employee skills and skill levels, but more advanced criteria, such as employee preferences, career objectives, or personalities, are not taken into account. Some services include custom fields that can potentially be used to implement some additional criteria and employee characteristics, but the effectiveness of such customisation is not guaranteed. In all the examined services, the skill-based search for employees is executed on a hit-or-miss basis, meaning that an exact match is expected. If no employee fills all the search criteria a manager has entered, no results are returned. Some services enable the search for employees that fill at least one of the search criteria but do not offer any insight on the relative ranking of the results. The target audience of the services is the management of an organisation. Few of them offer any interface for employees to fill and update their own details, such as skills and skill levels.

3 RESEARCH METHODOLOGY

Design science research (DSR) is applied as a research methodology in this thesis. DSR is first introduced based on a literature review, examining its phases and objectives. After the introduction, a detailed study design followed in this thesis is presented.

3.1 Design science research

The goal of design science research is to solve a problem by developing an artefact while providing sufficient rigour of the design process and evidence of the usefulness of the artefact [74]. A DSR publication should include enough technical information so that an implementation could be created with relative ease [75], while not forgetting the theoretical basis [74]. Compared to routine design, the DSR process is rigorous and creates new knowledge, whereas routine design utilises existing knowledge [74], [76], [77]. The knowledge created by DSR can later be used in routine design. DSR is used in many disciplines, such as architecture, engineering, education, health care, computer science and management [53], [76], [78]. In management disciplines, the goal of DSR is to advance the organisation to a desired direction by designing organisational artefacts, such as information and decision support systems [53].

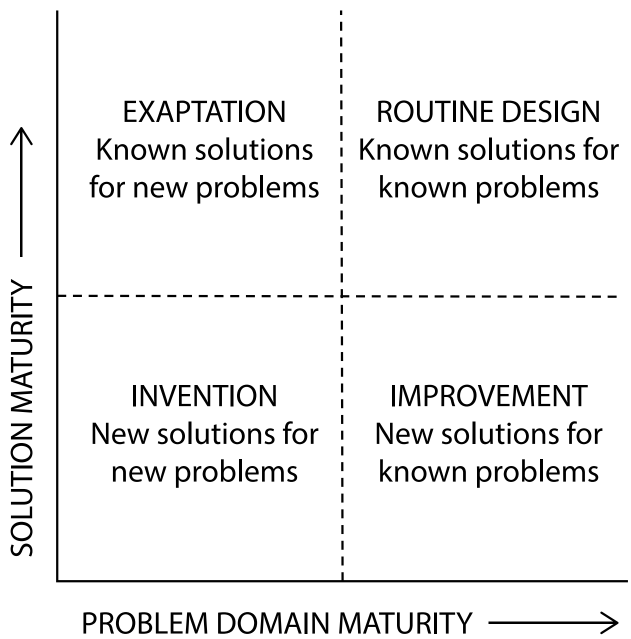


Figure 10. Design science research knowledge contribution framework. Adapted from [77].

Artefacts are human-made objects that are used to address specific problems [79]. Artefacts created by DSR are mainly constructs, models, methods and instantiations [74], [80], but can also be frameworks, architectures, design principles and design theories [78]. Design theories are the highest level of knowledge DSR can produce, whereas instantiations are usually context-specific, and so represent the lowest level of DSR contributions [77]. DSR artefacts are rarely isolated but rather interdependent objects [81]. For example, constructs and models are used to create instantiations for a specific context [79]. There are generally four types of DSR knowledge contributions, as seen in figure 10: routine design, exaptation, improvement and innovation [77]. The knowledge contribution of routine design is low and is not generally considered as design science research. A certain level of novelty is required from DSR artefacts in the form of improvements, exaptations and inventions [77]. There should also be some level of generality in DSR results, as an artefact that can only be applied to a specific problem does not provide much value for the research community [79], [81], [82]. The value of DSR largely depends on the utility of the artefact [81].

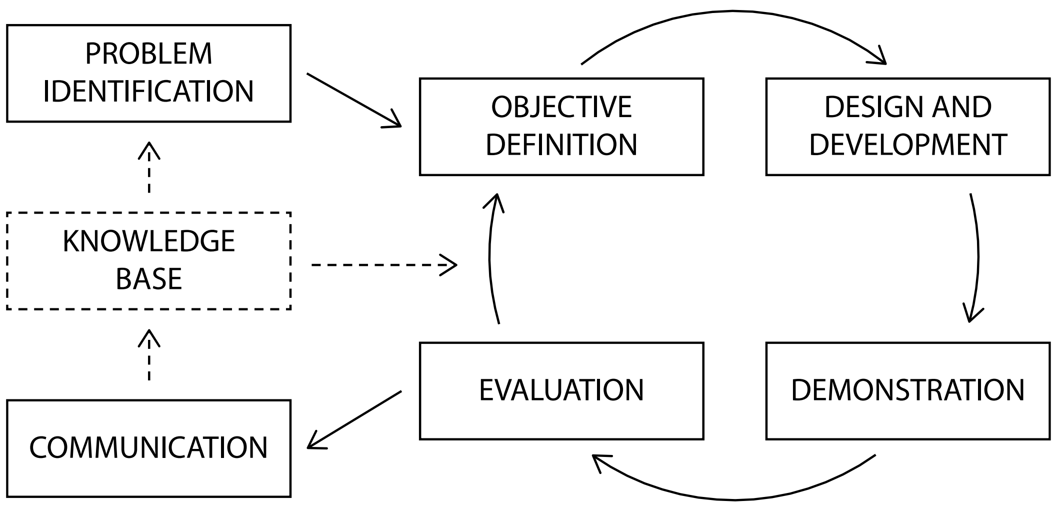


Figure 11. Design science research process model. Adapted from [78], [79], [83].

There have been many attempts to conceptualise DSR. In this thesis, the design science research methodology (DSRM) process, developed by Peffers, Tuunanen, Rothenberger and Chatterjee [83], will be utilised. DSRM identifies six distinct steps of DSR: problem identification and motivation, objective definition, design and development, demonstration, evaluation and communication. The order of the steps is not static, as the starting point of the research can, for example, be the evaluation of an existing artefact, instead of problem identification [83]. DSR, in general, is an iterative process [74], [83]. The knowledge gained during the first cycle can be used as input for the next cycle. Figure 11 describes the DSRM process, with the addition of knowledge base flow. The knowledge base consists of existing DSR knowledge and research methods [74], as well as knowledge from different disciplines related to the problem [84]. Effective DSR requires knowledge from the knowledge base so that the artefact requirements can be specified appropriately, and the rigour of the DSR process is guaranteed [74], [75], [84]. Compared to DSR, routine design poses an intellectual risk due to the decisions made based on inadequate knowledge [78]. Finally, when the results of DSR are communicated, new knowledge is added to the knowledge base so that it can be used in future research.

The goal of problem identification is to identify a problem that is unambiguous, well justified, and of general interest [79]. Problem identification requires knowledge of the problem domain so that the value of the resulting artefact can be justified [83]. In objective definition, both the objectives of the artefact and its requirements are defined [83]. The artefact is later evaluated against the defined objectives. Problem identification and objective definition require knowledge of the target context, for example, organisational knowledge, so that the resulting artefact has relevance in its intended environment [74], [79]. Different research methods, such as literature review, surveys, interviews and case studies, can be used in both activities [79], [85]. In the majority of DSR studies, relevant information is gathered from literature and different stakeholders in the problem domain [85].

In the design and development activity, an artefact is created based on its objectives and requirements [83]. The outputs may differ depending on the artefact type, for example, architecture and data model diagrams are essential parts of any instantiation, while not so relevant for methods. Related artefacts from research and practice should be analysed so that the novelty of the designed artefact can be justified [79]. The focus in the design and development phase is on the novelty of the design, rather than on the artefact construction [78]. In a sense, artefacts created by DSR are meta-artefacts, which create the foundation for the concrete artefact implementations [86]. Design principles, which are derived from identified design problems and their proposed solutions, are among the most valuable outcomes of DSR [77]. Decision made during the design process should be documented and justified so that they can later be added to the knowledge base [79].

Because design and development is a creative process, it is difficult to determine the right methods for each individual study [79], [86]. Different creative methods, such as brainstorming, emphatic thinking, sketching and prototyping can be utilised in the design process [79], while expert opinions are useful in validating the preliminary artefact design [84]. DSR is a collaborative endeavour by nature [85], [87], as design benefits from collective thinking and constant feedback loop. Different stakeholders should be involved from the beginning of the design and development process, so that early feedback is received, and the artefact design can be adjusted continuously.

After the artefact has been designed and developed, its functionality is demonstrated by applying it to the problem that it was designed to solve [83]. The feasibility of the artefact can be demonstrated through various research methods, for example, experimentations, simulations, and case studies [79], [83]. The following evaluation activity measures the efficiency, effectiveness, and impact of the artefact based on the pre-defined objectives and requirements [80]. Accurate metrics need to be developed so that the evaluation process is rigorous and credible. The metrics should be selected based on the goals of the artefact and may be qualitative, such as stakeholder satisfaction, or quantitative, such as system response time [83]. The evaluation activity can utilise the same research methods as the demonstration activity, as well as different analytical and testing methods [74], [77], [79]. As with the metrics, the evaluation methods should be selected based on the goals of the artefact [88]. For example, if the artefact solves a social problem, simulations are not useful in evaluating the feasibility of the artefact. Instead, the artefact should be studied in a real environment. The artefact can also be evaluated against existing related artefacts [79]. It is beneficial to evaluate the artefact at different stages of the design process using different evaluation methods, not only at the very end [88]. After evaluation, a new design cycle can be started with the added insight gained during the process [79], [83].

The final step of the DSR process is communication. DSR should be communicated so that it is useful for the research- and technology-oriented audiences, as well as the management-oriented audience [74], [83]. The research community may want to repeat the research, so a rigorous description of the research and the methods used is required. On the other hand, management needs sufficient information on which to base investment decisions if the artefact is to be used as a basis for a functional system.

3.2 Study design

Chapters 4–7 describe the design science research process carried out in this thesis. The structure follows the design science research methodology proposed by Peffers et al. Each phase in the methodology corresponds to a specific chapter in this thesis as described in table 2, demonstration and evaluation being addressed in a single chapter. Table 2 also mentions the research methods used in each phase.

Problem identification and objective definition are based on the conducted literature review. Additionally, a manager in the case organisation is interviewed to gain more context-specific information about the human resource allocation process. One possibility would have been to conduct an extensive case study to gain first-hand experience on the problem in its context. However, it would have been extremely challenging, considering the thesis schedule. Semi-structured interview was selected as the interviewing method since it allows for a more informal and open-ended conversation compared to a structured interview. Every organisation has a unique process for allocating employees to projects, so it is important not to lock down the questions beforehand and adapt the interview based on previous answers. The knowledge gained during the literature review is used to form the initial interview template. As the literature review results represent a typical human resource allocation setting, it is natural to expand on the identified issues from the perspective of the case organisation. The interview will be conducted remotely to conform with the distancing recommendations set due to the ongoing COVID-19 pandemic.

Table 2. Thesis structure in relation to the design science research methodology.

Design science research methodology phase	**Thesis chapter**	**Methods**
Problem identification	4 Problem definition	Literature review, semi-structured interview
Objective definition	5 Artefact requirements specification	Literature review, semi-structured interview
Design and development	6 Artefact design and development	Brainstorming, prototyping, expert opinion
Demonstration	7 Artefact evaluation	Simulation, expert opinion
Evaluation		

The design and development phase is purposefully kept quite open regarding the methods used, as design is a creative process. At least brainstorming and prototyping are used to lay out the foundation for the design, while expert opinions are used for validation and early feedback. Architecture, class, and sequence diagrams are utilised to describe the artefact on a technical level.

Finally, demonstration and evaluation are carried out as a simulation. As employee-specific data is scattered across different information systems of an organisation, it is not viably to try to connect all that data to the artefact in the scope of this thesis. Instead, simulated data is generated so that the artefact can be evaluated in an artificial context. A manager from the case organisation is consulted for an expert opinion as well. Based on the simulation and expert opinion, the artefact can be evaluated against the objectives set in previous phases.

4 PROBLEM DEFINITION

In this chapter, the problem context is set by introducing the case organisation and describing the human resource allocation problem the organisation is facing. A manager responsible for the allocation decisions was interviewed to gain insight into the environment and the problem itself. The interview was semi-structured and was based on the literature review and previous knowledge about the organisation.

4.1 Case organisation introduction

The case organisation is a medium-sized unit in a multinational software consulting company. The unit is located in Finland and has around 75 employees working in two separate offices. Typically, the case organisation executes 20–40 projects concurrently, ranging from 10 000 € to 1 000 000 € in budget. Depending on the size of the project, some project teams only consist of one employee, while others can have up to 10 employees. Because projects are, for the most part, external, some clients may add their own employees or other consultants to the team. Only less than 5 % of the projects are internal, in which the case organisation has total control over the project team. Typically, the duration of a project ranges from three to five months, but in extreme cases, a project can last only two weeks, or as long as five years.

The case organisation can be divided into five different functional departments: sales, marketing, project management, service and user experience design, and software design. The human resource allocation process is mainly focused on the three latter departments, as sales and marketing departments consist of a few employees and are highly self-organising. There are three project managers in the organisation, and thus they are managing a large number of projects concurrently, typically from 10 to 20 projects. Service designer also are assigned to many projects concurrently, usually up to 10, as their full capacity is rarely needed in one project. Majority of the organisation’s employees are software designers, who are typically assigned to 1–5 different projects concurrently. Even if a software designer is assigned to one project in full capacity, they may still have to do infrequent maintenance or development work in projects that are no longer in active development. Therefore, it is rare that a software designer would truly be assigned to only one project at a time.

4.2 Problem description

As the case organisation has been growing rapidly in recent years, management and sales department are starting to have difficulties in recognizing and identifying the talent the organisation possesses. It is relatively easy to memorise the skills and preferences of a handful of employees, but as the number of employees is nearing 100, the task becomes exponentially more difficult. The case organisation is still using a highly manual process in human resource allocation, that heavily relies on the memory of an individual manager. A manager, who is responsible for allocation decisions, gets inquiries from the sales department in order to fulfil particular project team requirements. The manager has to browse through employee CVs, inquire project managers and individual employees, and memorise previous experiences and employee preferences to determine if the company employs suitable people for a specific project. The current process is not only time-consuming, but also poses a risk in not identifying all the available talent, and disregarding employee preferences and career objectives. Employee preferences are regularly collected, but their utilisation is not always guaranteed due to the human-centric process. After the potential employees are identified, their availability has to be confirmed from an enterprise resource planning (ERP) system, as well as project managers and individual employees.

The case organisation has an ERP system that is used for project and human resource management tasks but is lacking in key human resource allocation features. Employees can input their skills in the system, but only by name. Skill levels are not included, which makes it impossible to search for a specific skill at a given level. Additionally, the system only allows searching for one skill at a time, making project team formation difficult. It is also not possible to allocate employees to projects preliminarily. The case organisation utilises Excel to track potential employee allocations, which adds another level of complexity to the process.

As a general observation, the major challenge in the human resource allocation process in the case organisation is information fragmentation. The essential information that is needed for allocation decisions is fragmented across different systems and people, making it challenging to make neutral and consistent decisions. It is also time-consuming for the manager to individually check every potential employee, as searching for project teams is not enabled by existing systems. The process involves a lot of communication between sales and the manager, who needs to communicate with project managers and employees. Human error is always present when communication is the primary tool used in a process.

5 ARTEFACT REQUIREMENTS SPECIFICATION

Based on the literature review and reaffirmed during the interview, a versatile decision support system would be ideal for helping with the human resource allocation problem in the case organisation. As information fragmentation is the major problem complicating the human resource allocation process, it was essential to bring all the key information into a single, easy-to-use decision support system.

It was clear from the start that the case organisation utilises nearly all of the factors identified in figure 8 when making the allocation decisions. The most critical factors were employee skills, capacity, and overall expenses. These factors are actively monitored, which makes sense because they are easily quantifiable. However, factors such as employee preferences, personality, productivity, and reliability also play a significant role in the allocation process. Especially personality, productivity, and reliability are difficult to measure, and it is up to the manager to utilise their own experience and knowledge of the employees to select the most suitable candidate for a given project in terms of personality. Because of the difficulty of measuring and quantitatively utilising the latter factors, it was decided to focus on more quantitative factors, such as skills and skill levels. After all, they are the primary factors that determine if an employee can be considered for a specific project or not.

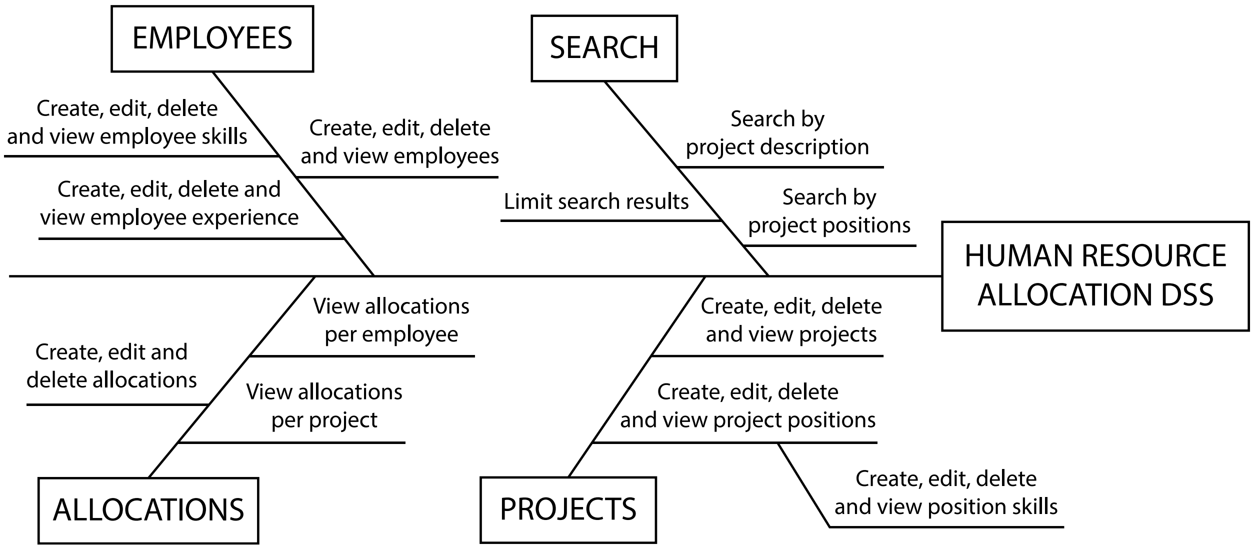


Figure 12. Artefact feature tree.

Desired features and requirements for the decision support system were discussed with the manager based on the literature review and previous knowledge of the case organisation. A summary of the features is presented in figure 12. To be able to search employees for a specific project, the system must enable the creation of projects. Different positions could be added to the project to achieve more relevant search results and keep the system versatile. For example, a project might have two different positions: a back-end developer, and a front-end developer. Back-end development requires a different skill set compared to front-end development, so, logically, the system should allow search for employees individually for each position.

The manager should have some control over the search method and results. The system should allow the manager to search employees for the whole project, or each project position individually. Some search result limitation would also be useful, so that, for example, the manager could examine the ten most suitable employees for a specific project. Date filtering was considered as well so that the system would search for the best candidates that are available during the intended project or position schedule. However, in some cases, it is possible to transfer employees between projects, so it was deemed that without date filtering, the manager would have more control over the allocation process.

The system also needs to enable the creation and storage of employees so that they can be considered in the search process. Only key information is recorded of each employee, such as name, skills, previous experience, and preferences, to keep the system simple. More sophisticated factors, such as personality, could have also been included in the system. However, because the case organisation does not monitor such information, it was decided to focus on the essential, quantitative factors in the scope of this thesis. Furthermore, it is subjectively easier for the manager to identify and memorise different personality types and reliability of different employees, as opposed to memorising what kind of skills they possess and in what kind of projects they have been working in the past. A decision support system is better in handling a large number of skills, while people still excel in evaluating human characteristics and different soft skills. However, when the employee count keeps growing, it may be worthwhile to include human factors in the decision support system. No manager can memorise hundreds of personalities, but less than a hundred is still manageable.

The system should also allow the planning of employee allocations to projects, to consider the project-based nature of the case organisation. After interpreting the results from the search functionality, employees could then be allocated to a specific project for a given period. It should be possible to examine each employee and their past, current, and future allocations. The system should alert of situations, where an employee is assigned to too many projects at the same time, to provide feedback for the manager. It should also be possible to allocate employees to projects preliminarily and indicate if the allocation has not yet been confirmed. This functionality would further reduce the information fragmentation present in the case organisation.

It is essential that employees are able to update their information so that a single manager is not responsible for keeping the skills and experience of employees up to date. The user interface should be easy to use so that employees can manage their data without any difficulties. It should also be possible for employees to examine their allocations, so that they can, for example, verify the capacity that has been allocated to a specific project. Figure 13 is a use case diagram containing the key features of the artefact. The artefact has two main user groups – managers and employees – and while managers have access to all the artefact features, employees can view information related to themselves.

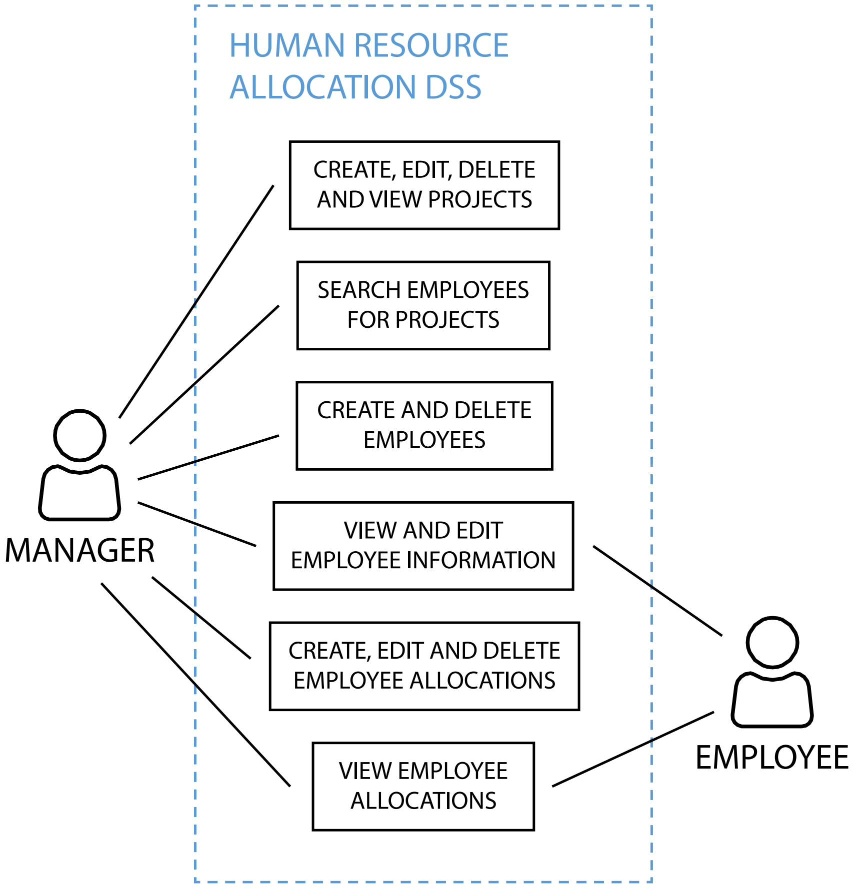


Figure 13. Artefact use case diagram.

As stated before, the case organisation has some of the required information already available in separate systems. No integrations will be built between the existing systems to keep the artefact at a certain level of generality. This way, the artefact can be applied to several different contexts and modified accordingly to fit the needs of a particular organisation. When considering production implementation, the effort needed to integrate the existing systems with the artefact should be evaluated. It would save time and effort not to begin the collection of the information from the start. Information is the key resource of the artefact, and the more accurate information the system has access to, the better.

Based on the requirements specification, the main objectives of the artefact are presented below. The objectives defined here are closely monitored during the artefact design and development phase. After the artefact development, the artefact can be evaluated against the defined objectives.

1. **Provide useful decision-making support for the human resource allocation process**
   The most crucial objective of the artefact is to provide useful insight and support for the manager who is making the allocation decisions. The manager should also have confidence in the artefact. The manager can evaluate this objective.
2. **Provide an easy-to-use interface for allocation decision-making and employee information maintenance**
   The artefact must have a straightforward and intuitive user interface to make an impact on the allocation process. The user interface should be easy to use for both the manager and the employee. The user interface can be evaluated by the manager and other potential users.
3. **Outperform existing solutions of the case organisation**
   For the organisation to consider implementation of the artefact, it has to outperform the existing solutions of the organisation. This objective can be evaluated when the artefact is compared to the existing solutions.
4. **Outperform existing commercial solutions**
   For the artefact to have an impact on the problem domain, it should also outperform the existing commercial solutions. This objective can be evaluated when the artefact is compared to the existing solutions. However, it is not viable to include every existing commercial human resource allocation solution or tool in the comparison. In the evaluation phase, solutions identified in the literature review will be used as a reference of the commercial state of the solution domain.

6 ARTEFACT DESIGN AND DEVELOPMENT

The design and development of the artefact was a creative process, and it was initiated after the artefact requirements and objectives had been formed. The very first step was to brainstorm what some of the core functionality could look like, and what kind of data models would be required. Initial wireframes of the artefact can be seen in appendix 1, which portray the key features listed in figure 12. The wireframes could later in the design and development process be used as reference points of the desired outcome and helped summarise the general idea behind the artefact.

In the beginning, it was concluded that the artefact would have two main sections: project and employee management. Project management allows managers to create, update, delete, and view projects. When a specific project is selected, it can be supplemented with positions. Allocations can also be created for projects, which are then reflected in the employee section. The search functionality is linked to individual projects. This way, managers can quickly search for suitable employees for either specific positions, or the project overall. The employee section lists all the employees in the organisation and allows managers to handle them in the same way as projects. Also, employees themselves can utilise the employee section to update and view their information. Each employee has preferences, skills, and experience. Also, allocations can be examined on the personal page of a specific employee.

After the main idea behind the artefact had been formed, the overall architecture of the artefact was designed. The architecture diagram is described in figure 14, and as can be noticed, it closely resembles the general decision support system architecture described in figure 9. The architecture consists of three main layers: front-end, back-end and database. The front-end layer is visible for the user and contains the actual user interface and an application programming interface (API) client. Users interact with the user interface, which provides access to the back-end via the API client. The API client allows the exchange of data between the front- and back-end. For example, to create a new project, the user inputs all the necessary information in the user interface, which is then transferred to the back-end and saved in the database. The back-end handles all the business logic of the artefact and the database access. The back-end contains the API, which relays all the different requests from the front-end to the rest of the back-end. Some requests require business logic, such as the search functionality, while other requests can straight fetch information from the database. The back-end contains a data access layer to simplify database operations. The data access layer handles all the different database operations and offers the rest of the back-end an easy interface to fetch, create, update, and delete information.

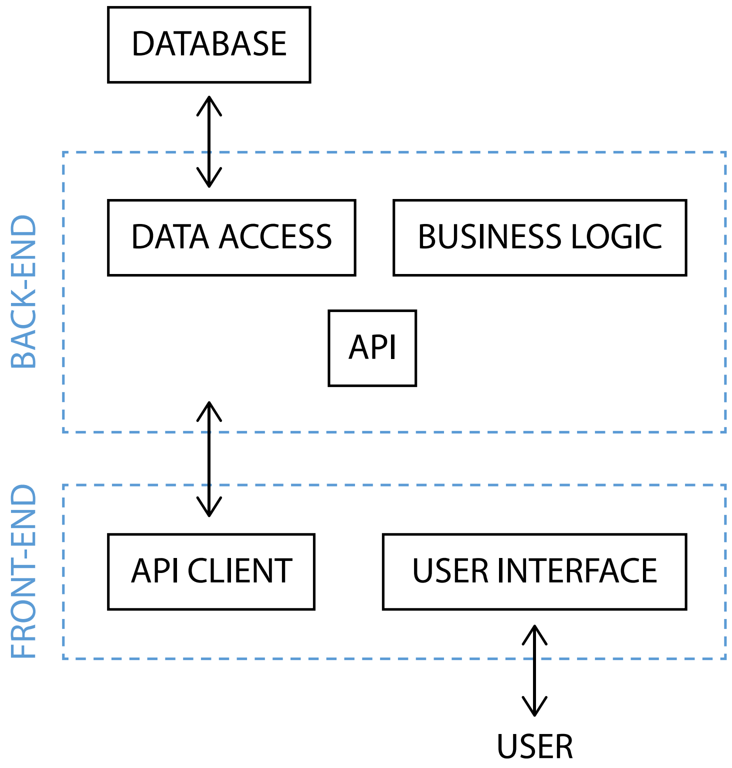


Figure 14. Artefact architecture diagram.

Technologies used in the artefact development were decided based on the architecture design and artefact requirements. List of the selected technologies can be examined in table 3. In general, the latest version of each framework or library was used. PostgreSQL was selected as the database technology, as it is versatile and fast, and a relational database fits the artefact needs well. PostgreSQL is also widely supported, so other technologies can easily be integrated with it. The back-end is based on Node.js, which is a popular framework with extensive support and an active community. Like PostgreSQL, Node.js is very versatile and can be extended with numerous open-source libraries, so it is an easy framework to build various systems on. Prisma was selected to handle data access between the database and the back-end. Some of the business logic utilises QMiner, which is a data analytics platform. Because QMiner is not in active development anymore, it does not support the latest version of Node.js. Therefore, a previous version of Node.js was used. QMiner is later discussed in more detail with the employee search functionality. The API utilises graphql-yoga, which is based on a popular Node.js web application framework, Express. graphql-yoga enables support for a GraphQL API, which the artefact utilises in data exchange between the front- and back-end. Compared to the traditionally used representational state transfer (REST), GraphQL offers more flexibility by allowing the front-end largely dictate in which form the data is needed. This approach does not require as much synchronous development between the API and the API client after the initial data models have been confirmed. If the front-end would suddenly need some additional data from the back-end, the modification only needs to be done in the API client, given that the data exists and is accessible in the first place.

Table 3. List of utilised technologies.

Layer	Component	Technology	Version
Database		PostgreSQL	12.3
Back-end	Base framework	Node.js	10.22.0
	Data access	Prisma	2.1.3
	Business logic	QMiner	9.2.4
	API	graphql-yoga	1.18.3
Front-end	Base framework	React	16.13.1
	API client	Apollo Client	3.0.0
	User interface framework	Material-UI	4.11.0

The front-end is based on React, which is a popular framework used to build web user interfaces. As well as Node.js, React has an active community, and therefore, various libraries exist that can augment React and ease the user interface development process. Apollo Client is used as the API client, which handles all the data exchange with the back-end. In addition to processing network requests and responses, Apollo Client also offers cache management for the front-end. Caching the responses reduces network traffic, as the API client does not need to fetch data again if it has already been fetched and no modifications have been made. Additionally, Material-UI, which is based on the Material design system, is used to provide basic user interface elements. A user interface framework drastically speeds up the user interface development, as most of the effort can be spent on the development of the actual artefact functionality.

The data models of the artefact were revised multiple times during the design and development process. The final class diagram is presented in figure 15. The project type has one-to-many relationship with project positions, meaning that a project can have multiple positions, but a position only belongs to one project. Similarly, project position can have multiple position skills. In addition to name and level properties, a position skill can be compulsory, meaning that only employees possessing that skill can be considered for the position. Employees can have many skills, as well. Also, employees can have multiple experience objects. The employee experience can be seen as personal project history, describing what kind of projects the employee has been working on, for which customers, at which position, and using what kind of skills. The search functionality can utilise all of this information. Projects and employees can have multiple allocations, which are used to visualise the capacity status of each employee. Allocations can also be created as drafts, helping the managers to perceive the overall status of an employee.

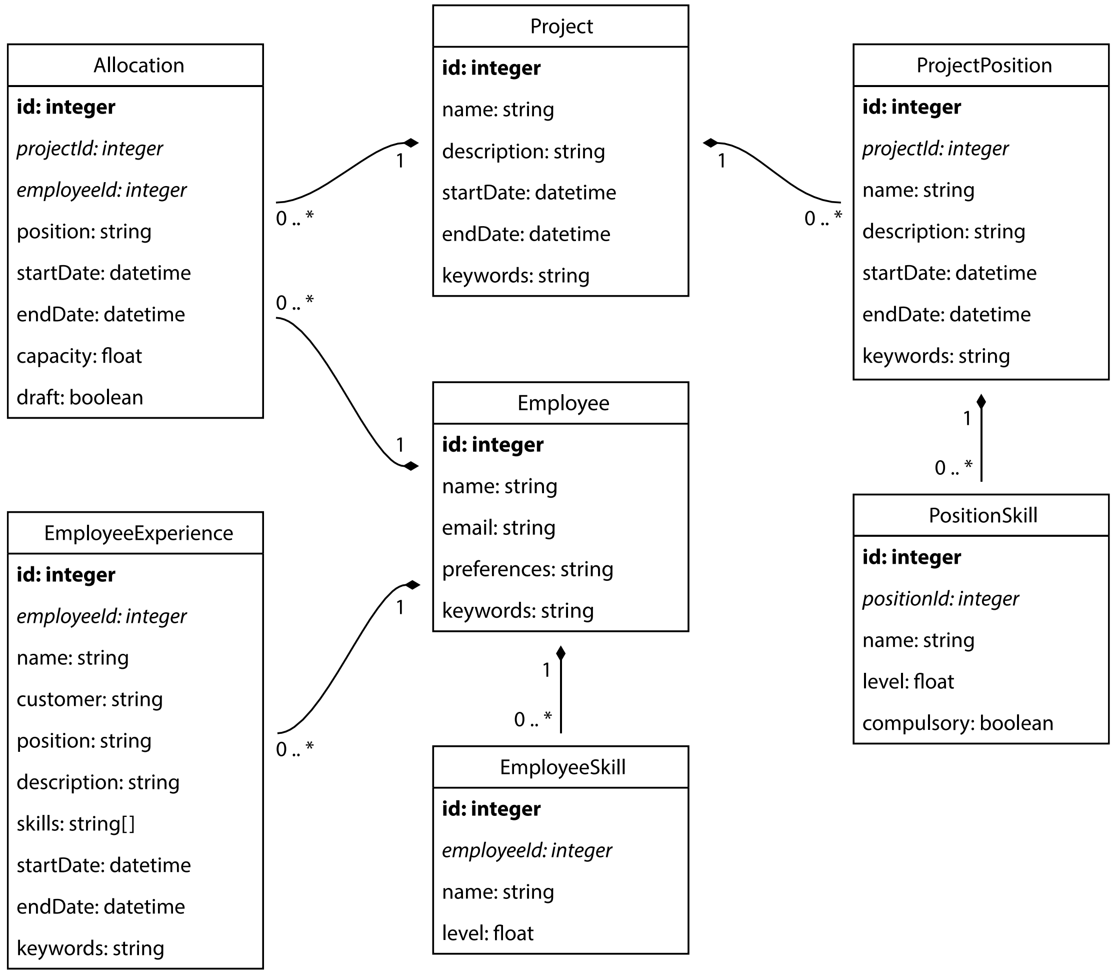


Figure 15. Artefact class diagram.

The employee search functionality is based on keywords, which are generated for projects, project positions, and employees. Instead of searching the database for specific skills and skill levels, employee keywords are compared with project or project position keywords. Employees are then scored based on how similar the keywords are. Keyword comparison allows for much more versatile search with relatively low implementation effort. It would be very laborious to manually include every desired employee or project property in database queries. This kind of approach would result in complex logic that would be difficult to maintain and troubleshoot. On the other hand, keyword comparison is straightforward and enables the comparison of numerous properties without significant maintenance concerns. It is also easy to add new properties to the search function or remove existing ones since the operations only require modifications to the keyword formation. Another advantage of keyword comparison is the search performance. Because keywords are formed at the time an object is created or modified, all the keywords for different employees and project positions will already be ready when a manager wants to initiate employee search. If the properties were queried individually from the database, improving the performance would be extremely difficult, especially when the system has to sift through hundreds of employees.

Table 4. Keyword formation.

Entity	Property	Coefficient
Project	Description	1
	Name	1
Project position	Name	4
	Description	1
	Skills	Skill level × 4
Employee	Preferences	1
	Skills	Skill level × 4
	Experience	
Employee experience	Name	1
	Customer	4
	Position	4
	Description	1
	Skills	4

Keywords are formed based on different properties depending on the data type. The complete keyword formation is explained in table 4. Project keywords consist of only the project name and description. The keywords of a project position include the position name, description, and skills. Employee keywords consist of preferences, skills, and all the keywords of each experience. Employee experience keywords include the experience name, customer, position, description, and skills. Some properties are multiplied to highlight their importance and provide better search results. For example, skills are multiplied based on the skill level and a constant coefficient. If an employee has a high skill level in a specific skill, or if a position requires a high level of some skill, the skill name is multiplied to highlight the importance. Additionally, project position name, the customer, and position name of an employee experience are multiplied by a constant coefficient. Positions names tend to be similar across different kinds of projects, so it is useful to highlight them in keywords. It is also useful if an employee has experience with a specific customer, so the property is also highlighted. The benefit of highlighting selected properties is that if employee keywords contain high amounts of specific words, they are likely to score higher if project or position keywords contain equally high amounts of the same words. For example, if a position requires some skill with a high level, employees possessing high level on that skill will be emphasised in the search results. The static coefficient used in the artefact development was “4”. The value was selected based on testing because it seemed to yield the best results on the test data. However, with drastically different data, some other coefficient might work better. The effect of the coefficient also depends on the actual keyword comparison implementation.

Before keywords are saved, the data is normalised to provide more accurate comparison, and therefore better search results. Firstly, any punctuation and extra whitespace is removed. Secondly, the most common function words are removed. Function words are words that have little meaning or are used for grammatical purposes only. For example, conjunctions, such as “and”, “or”, and “but” are removed. Normalisation cleans up the descriptive data, such as project, position, and employee experience descriptions. Normalising the keywords reduces the probability of false positives and improves the overall search results, as the comparison process can focus on the keywords that actually have meaning.

The keyword comparison was implemented utilising QMiner, which is a data analytics platform written in C++ [89]. QMiner offers a range of machine learning and statistical tools, especially for text mining purposes. QMiner has a JavaScript API, which makes it possible to utilise it in Node.js projects. The keywords of all the employees are imported to the database implementation of QMiner to begin the keyword comparison process. In case employees are searched for a position that has compulsory skills, only keywords of employees possessing those skills are imported. Next, a text feature extractor is applied for each keyword record. The feature extractor prepares the keywords for data analysis by mapping them to linear algebra vectors. Finally, linear algebra is applied to the matrix containing all the employee keyword vectors, and the vector containing the project or project position keywords. This process yields a score for each employee, describing the similarity with the target keywords. The employees can then be sorted based on the similarity score and displayed to the user. It can be argued that the inclusion of the text mining tool makes the artefact an intelligent decision support system. Text mining allows the inclusion of more diverse allocation factors, such as employee preferences, and therefore provides more accurate results for the manager.

A general project creation flow is described as a sequence diagram in figure 16. The manager initiates the creation process by inputting all the required information in the user interface. The data is then transferred to the back-end, where it is stored in the database. Before saving the data, keywords are generated for the project. All the other creation and modification mostly follow the same pattern. However, if an object does not require keywords, the business logic layer is bypassed.

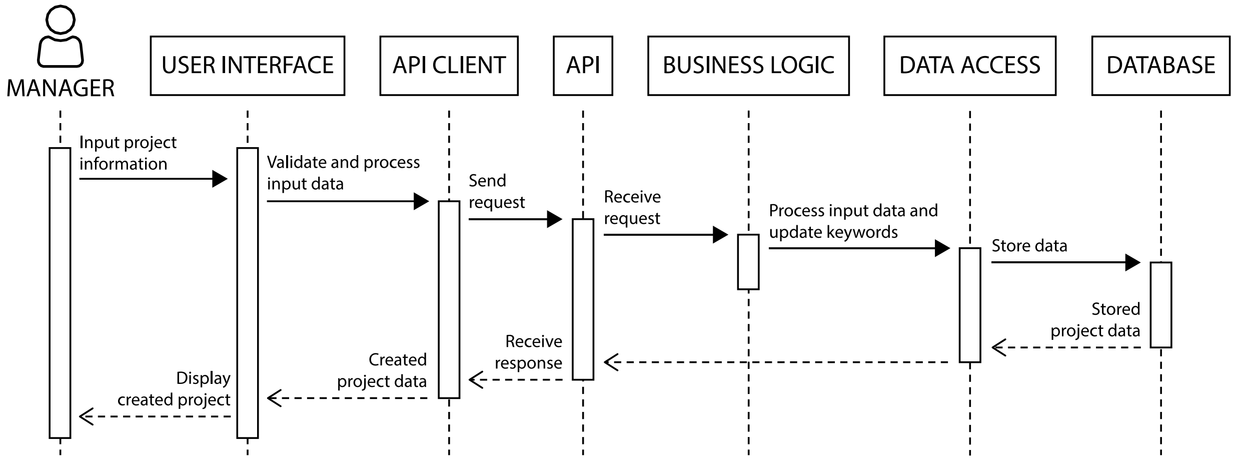


Figure 16. Sequence diagram of project creation.

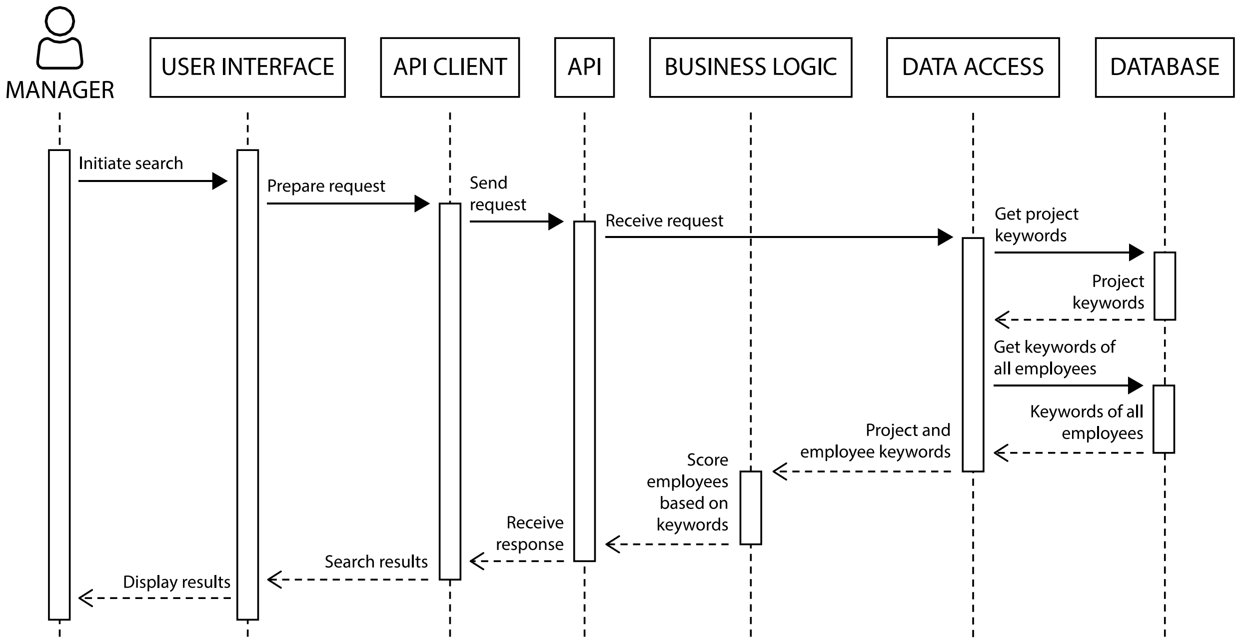


Figure 17. Sequence diagram of employee search by project data.

Figure 17 describes the employee search flow based on project data only. Keywords of the project and all of the employees are first fetched from the database. QMiner is then utilised in the business logic layer to process the keywords and score the employees. Figure 18 describes the search flow based on the project positions. The process is mostly similar, but each position is processed individually. Also, only employees who fulfil the compulsory skill requirements are considered. The results are grouped by project position, so the manager can easily compare the results individually for each position.

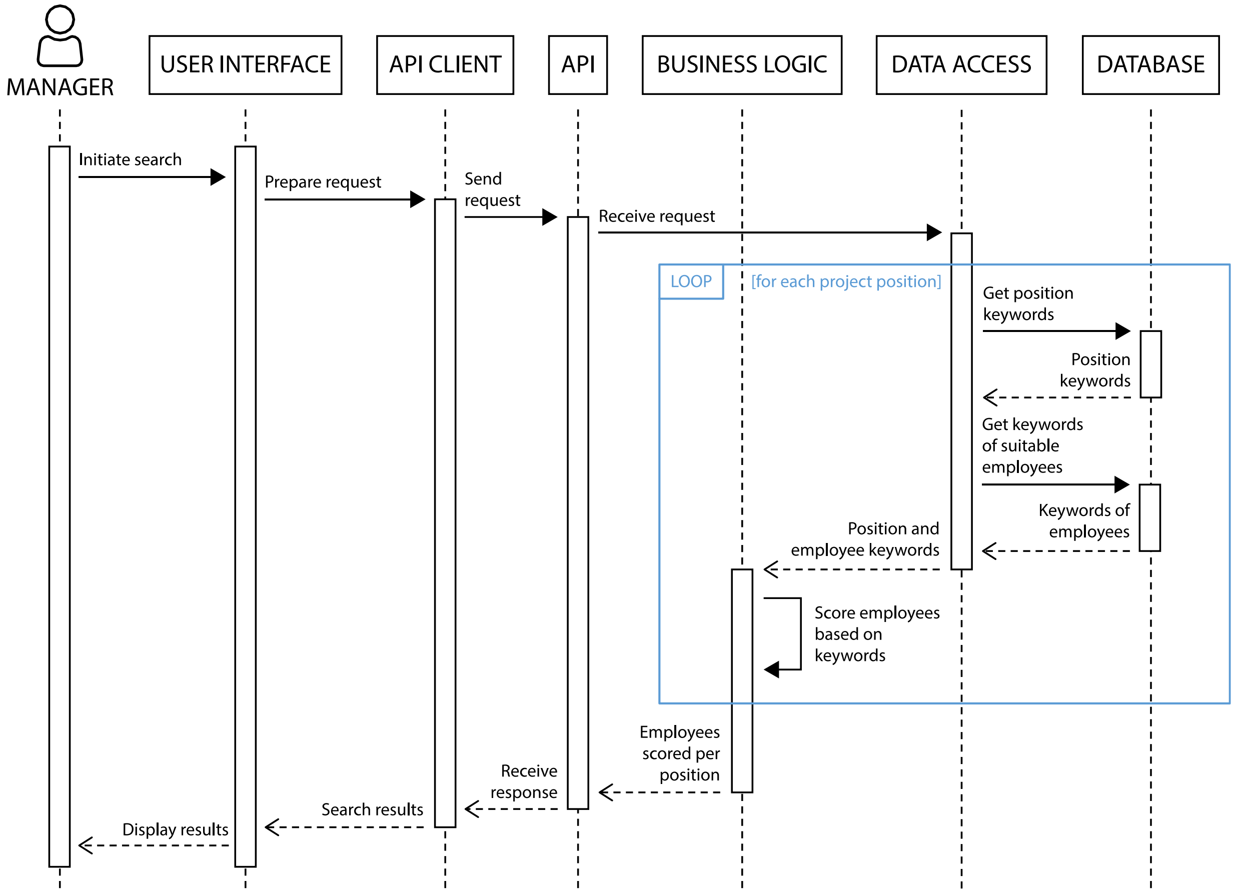


Figure 18. Sequence diagram of employee search by project positions.

Screenshots of the finished artefact are presented in appendix 2. The number of projects and employees is low in the screenshots to keep the layout of this document clean. Something to note is the calendar component in the employee view, which displays the past, current and future allocations of an employee. Draft allocations are coloured differently from confirmed allocations so that the manager can use the tool as support for planning. The component also highlights if an employee has too high, or too low allocated capacity. In general, the resulting artefact fulfils the requirements well and offers an easy-to-use and logical user interface. The source code of the artefact is available on GitHub[1].

[1] https://github.com/ilarisahi/hra-dss

7 ARTEFACT EVALUATION

After the artefact design was completed and developed, the artefact was demonstrated to the case organisation. Because no integrations were utilised in the artefact development to uphold general applicability, simulated data was used. The data was created manually to ensure that the data was realistic and relevant to the case organisation. 30 employees were created with different skill sets and skill profiles. For example, developers with strong front-end experience have a different kind of skill set compared to back-end developers. On the other hand, full-stack developer profiles have experience in both front- and back-end technologies. Service and user experience designers were also created, focusing on the experience in different design tools. Additionally, two employees with cloud service experience were included in the simulated data set. Each employee had varying skill levels to simulate variance, and some experience objects to complement the skill profile. Employee preferences were also simulated. Only two projects were created since one project already can have multiple different positions that trigger the search algorithm individually. One of the projects only had a description, while the other had five different positions: front-end developer, back-end developer, full-stack developer, cloud engineer, and user experience designer. The positions were manually created with matching description and a medium level of related skills.

The demonstration was organised remotely due to the ongoing social distancing recommendations. The manager responsible for the human resource allocation decisions, and two other employees of the case organisations participated in the meeting. In short, all the key artefact features were demonstrated with an emphasis on the employee search and allocation functionality. The architecture and business logic were also introduced to give the participants some background information on the artefact. The artefact was received well and got positive feedback. Immediately it sparked a conversation on how the artefact could be integrated into the existing systems of the case organisation.

Evaluation of the artefact was conducted based on the four objectives set earlier in the design science research process. Evaluation mostly took place in parallel with the artefact demonstration as the same simulated data was used during both phases. The primary objective of the artefact was to provide useful decision-making support for the human resource allocation process. This was subjectively evaluated by the manager of the case organisation. The verdict was that the artefact would greatly help in supporting the allocation decisions if it was integrated with other related systems of the organisation. Because some of the data is already present in the organisation, such as confirmed allocations, basic employee information, and project descriptions, it would be redundant to maintain the same information in the artefact. Integration potential was considered during the artefact design process. Therefore the database and input forms in the user interface are loosely coupled with the core business logic of the artefact. It would be relatively easy to integrate the artefact with any system, of course, depending on what kind of external interfaces the other systems offer. The accuracy of the search functionality was perceived as impressive and fast. Also, the employee allocation calendar component was thought to be very useful since it allows the planning of multiple preliminary allocations in the project-based environment. Some questions were raised on the date constraints that can be set for projects and project positions. It was a conscious decision not to impose strict date filtering since the case organisation often can fine-tune existing allocations. However, some companies might want to restrict the search results more strictly regarding the date constraints.

The second objective of the artefact was to provide an easy-to-use user interface to maintain employee information and find suitable employees for projects and specific project positions. The user interface was evaluated during the artefact demonstration, and the consensus was positive. The overall flow of the artefact seemed logical, and all the key components were present. One of the valuable outputs of this design science research is the identification of key components required in a system that is designed to support the human resource allocation decision-making in a project-based organisation. One useful addition could be to include an overall calendar view displaying allocations of all the employees. The manager could speed up the planning process if all the allocations of resulting employees could be displayed at once.

It was also crucial that the artefact would outperform the existing systems used by the case organisation in the allocation process. This can be objectively evaluated to be the result. In the ERP system, the case organisation can search for employees based on a single skill name, which is not so useful when employees are searched for a position requiring multiple skills at different levels. The artefact also had the objective to outperform existing commercial systems that provide human resource allocation support. It is more challenging to comprehensively evaluate the artefact against the commercial solutions, due to the sheer number of them. It can be said that the artefact mostly outperforms the systems, considering the ones identified during the literature review in chapter 2.5. Most of the identified systems only offer a basic search of employees based on skills, and in some cases, skill levels. However, the functionalities tend not to be very versatile. If an employee only partially possesses the queried skills, or at a wrong level, they are excluded from the results. Also, none of the systems enables the search for an entire team at once. The keyword matching developed in this artefact seems to be unique to the solution domain and gives it the edge over the existing commercial solutions. When considering the planning capabilities for project-based organisations though, in some cases, the artefact falls short. Some commercial solutions offer short-term planning support, which the artefact does not provide. Also, in many commercial solutions, it is easier to get the overall picture of the allocation status of multiple employees at once. The artefact is mostly meant for medium-term human resource allocation planning support but could also be extended to support long-term and short-term planning. In general, the artefact outperforms the existing commercial solutions on the employee search functionality but does not offer the same level of planning support as some other systems.

8 DISCUSSION

The results of the artefact evaluation show that the artefact design and development was mostly successful. The only objective that was not quite reached was the outperformance of existing commercial solutions. The planning capabilities of the artefact are somewhat lacking compared to some of the commercial solutions available. However, this thesis has multiple contributions to the solution domain of the human resource allocation problem. This chapter discusses the contributions of this thesis, the managerial implications, the validity and limitations of the research, and proposes some topics for future research.

8.1 Contributions

This thesis makes multiple contributions to the human resource allocation knowledge base. Firstly, factors influencing human resource allocation decisions in a project-based organisation were comprehensively identified and summarised in figure 8. They were summarised from multiple different sources that all provided a partial set of factors. The factors are a good starting point for any organisation that has some kind of a human resource allocation process and wishes to improve it. The list of factors can act as a checklist which the organisation can prioritise to suit their needs.

Also, a human resource allocation decision support system design is proposed. The design includes the system architecture, data model, and key user interface components. The source code has also been made publicly available. Past research has almost unanimously focused on the algorithms behind the employee search, with little to no inclusion of user interface design. The DSS proposed in this thesis identifies the key components required for effective human resource allocation, which makes it easier for organisations to implement similar systems.

This thesis also suggests the utilisation different intelligent techniques to support the allocation decision-making, particularly text mining. Using keywords to match employees with projects and project positions makes the system versatile and easy to maintain. New factors can be included in the matching process with relative ease. The suggested method performs well and offers accurate results for the managers. If the artefact would be classified according to figure 10, it is something between an improvement and an exaptation. On the one hand, the artefact improves the search functionality of entire project teams compared to previous research. On the other hand, the artefact introduces text mining to help in the human resource allocation, which has not previously been utilised.

8.2 Managerial implications

The potential of a decision support system to aid in the human resource allocation process is remarkable. Information fragmentation makes it challenging to match employees with projects. As the employee count of the organisation increases, it becomes more difficult for managers to memorise the skill set of each employee. It is also challenging to make unbiased decisions because not all preferences of all employees can be manually included in the allocation process. Some commercial solutions offer a very basic level of allocation decision support in the form of employee search by individual skill and skill level.

The DSS proposed in this thesis can comprehensively include multiple different factors in the allocation process based on the organisational strategy and needs. The system is cost-effective to implement and maintain and performs accurately. It can also be used as a medium-term planning tool, as each employee has a visualised status of their current and future allocations. However, it should be carefully determined if the organisation would benefit from a DSS in the human resource allocation process.

8.3 Research validity and limitations

The design science research conducted in this thesis is firmly based on literature, which strengthens the validity of the results. Previous studies were explored comprehensively, and multiple different sources were utilised. The case organisation was also examined thoroughly through interviews and multiple years of personal experience. Human resource allocation is a global problem that is present in many different industries. Therefore, the artefact designed and developed in this thesis was consciously kept at a fairly general level, so that the results could be applied in a wide range of contexts. For example, no integrations to specific systems were included, and the different components of the artefact are loosely coupled. It is easy to modify the artefact design to fit several different use cases depending on the type of data the organisation has and collects and the systems the organisation has in use.

Some limitations remain in the research. For example, the effects of different language were not included in this study. The text mining technique used in matching employees with projects may not work as well with some other languages, as it works with English. English is a straightforward language to map common words with two different sets of words, as meaningful words generally have only one grammatical case. On the other hand, in Finnish, for example, words have many different grammatical cases depending on the context and placement of the word. Therefore, finding keywords from free form text, such as employee preferences, might not work as well.

8.4 Future research

There is still a lot of demand for human resource allocation research from different point of views. First of all, the inclusion of more complex factors, such as productivity and personality, could be further studied. There are virtually no studies that thoroughly examine the collection and maintenance of such factors. Additionally, it is still unclear whether all organisations can benefit from human resource allocation decision support systems, or if an organisation needs to reach a certain size for it to be beneficial. Some industries might be better suited to utilise such tools, while others do not need complex support in the process. Utilisation of intelligent methods in the allocation process is also a relatively new topic with many unanswered questions. For example, the potential of machine learning could uncover novel methods that can aid managers to make better decisions.

The artefact designed and developed in this thesis could also benefit from further development. The artefact could be enhanced with better planning tools to extend the utility of the artefact to long- and short-term human resource allocation planning.

9 SUMMARY

This thesis studied the human resource allocation problem in the context of a project-based organisation. When organisations grow, it becomes more and more difficult for the managers to identify all the available talent in the organisation and all the different preferences of individual employees. Therefore, a decision support system, that can sift through large amounts of data and consistently support the managers in the allocation process is in high demand. In this thesis, such a system was designed for a case organisation operating in the software consulting industry.

The knowledge base of the allocation problem was studied extensively during a literature review. The problem was mainly approached from the point of views of software engineering and project-based organisations. One of the most important contributions of the literature review is the identification of factors that influence the human resource allocation decision in a project-based organisation. Existing decision support systems for the allocation problem were also studied. It was concluded that previous research has mainly focused on the algorithms that match employees with projects, while commercial solutions offer a very basic level of human resource allocation support.

Design science research was utilised to design a decision support system that would improve upon both the previous research and commercial solutions available. The focus was on making an effective decision support system that users could interact with via a logical user interface. Text mining was used to match employees with projects by forming keywords of each relevant object. For example, keywords of an employee can be compared with the keywords of a project position. This comparison calculates a score for each employee, that can then be used to sort the employees based on the suitability. Employees, projects, and allocations can be managed via an easy-to-use user interface. Employees can maintain their information, while managers are able to manage projects and allocations. Managers can also use the user interface to search suitable employees for projects or specific project positions.

The case organisation recognised the value that the artefact could bring to the organisation. Employee suggestions provided by the search functionality were accurate, and the allocation planning tool would allow for a more coherent human resource allocation process. However, to maximise the utility, the artefact should be integrated with other information systems in the organisation. Because some of the data the artefact utilises is already stored and managed in other systems, it would be inefficient to also collect that information manually in the artefact.

REFERENCES

[1] Project Management Institute, *A guide to the project management body of knowledge (PMBOK® guide)*, 6th ed. Newtown Square, PA, US: Project Management Institute, 2017.

[2] J. Heagney, *Fundamentals of Project Management*, 4th ed. New York, NY, US: American Management Association, 2011.

[3] J. H. Payne, “Management of multiple simultaneous projects: a state-of-the-art review,” *International Journal of Project Management*, vol. 13, no. 3, pp. 163–168, 1995, doi: 10.1016/0263-7863(94)00019-9.

[4] J. Pfeffer and G. R. Salancik, *The External Control of Organizations: A Resource Dependence Perspective*, 1st ed. Stanford, CA, US: Stanford University Press, 2003.

[5] C. Stylianou and A. S. Andreou, “Human Resource Allocation and Scheduling for Software Project Management,” in *Software Project Management in a Changing World*, 1st ed., G. Ruhe and C. Wohlin, Eds. Heidelberg, Germany: Springer, 2014, pp. 73–106.

[6] M. C. J. Caniëls and R. J. J. M. Bakens, “The effects of Project Management Information Systems on decision making in a multi project environment,” *International Journal of Project Management*, vol. 30, no. 2, pp. 162–175, 2012, doi: 10.1016/j.ijproman.2011.05.005.

[7] Project Management Institute, *Software extension to the PMBOK® guide*, 5th ed. Newtown Square, PA, US: Project Management Institute, 2013.

[8] H. Kerzner, *Project management: a systems approach to planning, scheduling, and controlling*, 12th ed. Hoboken, NJ, US: John Wiley & Sons, 2017.

[9] IEEE Computer Society, *Guide to the Software Engineering Body of Knowledge*, Version 3. Los Alamitos, CA, US: IEEE Computer Society Press, 2014.

[10] M. Engwall, “No project is an island: linking projects to history and context,” *Research Policy*, vol. 32, no. 5, pp. 789–808, 2003, doi: 10.1016/S0048-7333(02)00088-4.

[11] G. Ruhe and C. Wohlin, “Software Project Management: Setting the Context,” in *Software Project Management in a Changing World*, 1st ed., G. Ruhe and C. Wohlin, Eds. Heidelberg, Germany: Springer, 2014, pp. 1–24.

[12] F. P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, 2st ed. Boston, MA, USA: Addison-Wesley Longman, 1995.

[13] Z. Laslo and A. I. Goldberg, “Resource allocation under uncertainty in a multi-project matrix environment: Is organizational conflict inevitable?,” *International Journal of Project Management*, vol. 26, no. 8, pp. 773–788, 2008, doi: 10.1016/j.ijproman.2007.10.003.

[14] D. Dalcher, “Rethinking Success in Software Projects: Looking Beyond the Failure Factors,” in *Software Project Management in a Changing World*, 1st ed., G. Ruhe and C. Wohlin, Eds. Heidelberg, Germany: Springer, 2014, pp. 27–49.

[15] C. Biesenthal and R. Wilden, “Multi-level project governance: Trends and opportunities,” *International Journal of Project Management*, vol. 32, no. 8, pp. 1291–1308, 2014, doi: 10.1016/j.ijproman.2014.06.005.

[16] R. Müller, *Project Governance*, 1st ed. Farnham, UK: Gower, 2009.

[17] E. G. Too and P. Weaver, “The management of project management: A conceptual framework for project governance,” *International Journal of Project Management*, vol. 32, no. 8, pp. 1382–1394, 2014, doi: 10.1016/j.ijproman.2013.07.006.

[18] Project Management Institute, *A guide to the project management body of knowledge (PMBOK® guide)*, 5th ed. Newtown Square, PA, US: Project Management Institute, 2013.

[19] M. Hendriks, B. Voeten, and L. Kroep, “Human resource allocation in a multi-project R&D environment,” *International Journal of Project Management*, vol. 17, no. 3, pp. 181–188, 1999, doi: 10.1016/S0263-7863(98)00026-X.

[20] J. R. Turner, “The management of the project-based organization: A personal reflection,” *International Journal of Project Management*, vol. 36, no. 1, pp. 231–240, 2018, doi: 10.1016/j.ijproman.2017.08.002.

[21] R. Gareis and M. Huemann, “Project management competences in the project-oriented organization,” in *People in Project Management*, 1st ed., J. R. Turner, Ed. Aldershot, UK: Gower, 2003, pp. 31–43.

[22] A. Keegan, R. Turner, and M. Huemann, “Managing Human Resources in the Project-based Organization,” in *Gower Handbook of Project Management*, 4th ed., J. R. Turner, Ed. Aldershot, UK: Gower, 2007, pp. 649–676.

[23] M. Miterev, M. Mancini, and R. Turner, “Towards a design for the project-based organization,” *International Journal of Project Management*, vol. 35, no. 3, pp. 479–491, 2017, doi: 10.1016/j.ijproman.2016.12.007.

[24] T. DeMarco and T. Lister, *Peopleware: Productive Projects and Teams*, 3rd ed. Boston, MA, USA: Addison-Wesley, 2013.

[25] P. Canonico and J. Söderlund, “Getting control of multi-project organizations: Combining contingent control mechanisms,” *International Journal of Project Management*, vol. 28, no. 8, pp. 796–806, 2010, doi: 10.1016/j.ijproman.2010.05.005.

[26] I. Hyväri, "Project management effectiveness in project-oriented business organizations," *International Journal of Project Management*, vol. 24, no. 3, pp. 216–225, 2006, doi: 10.1016/j.ijproman.2005.09.001.

[27] A. Platje and H. Seidel, "Project and portfolio planning cycle: Project-based management for the multiproject challenge," 1994. doi: 10.1016/0263-7863(94)90016-7.

[28] H. G. Gemünden, P. Lehner, and A. Kock, "The project-oriented organization and its contribution to innovation," *International Journal of Project Management*, vol. 36, no. 1, pp. 147–160, 2018, doi: 10.1016/j.ijproman.2017.07.009.

[29] M. Huemann, A. Keegan, and J. R. Turner, "Human resource management in the project-oriented company: A review," *International Journal of Project Management*, vol. 25, no. 3, pp. 315–323, 2007, doi: 10.1016/j.ijproman.2006.10.001.

[30] M. Huemann, "Considering Human Resource Management when developing a project-oriented company: Case study of a telecommunication company," *International Journal of Project Management*, vol. 28, no. 4, pp. 361–369, 2010, doi: 10.1016/j.ijproman.2010.02.008.

[31] M. Formentini and P. Romano, "Using value analysis to support knowledge transfer in the multi-project setting," *International Journal of Production Economics*, vol. 131, no. 2, pp. 545–560, 2011, doi: 10.1016/j.ijpe.2011.01.023.

[32] D. C. C. Peixoto, G. R. Mateus, and R. F. Resende, "The issues of solving staffing and scheduling problems in software development projects," in *Proceedings - International Computer Software and Applications Conference*, 2014, pp. 1–10, doi: 10.1109/COMPSAC.2014.96.

[33] P. Patanakul and D. Milosevic, "The effectiveness in managing a group of multiple projects: Factors of influence and measurement criteria," *International Journal of Project Management*, vol. 27, no. 3, pp. 216–233, 2009, doi: 10.1016/j.ijproman.2008.03.001.

[34] M. Yoshimura, Y. Fujimi, K. Izui, and S. Nishiwaki, "Decision-making support system for human resource allocation in product development projects," *International Journal of Production Research*, vol. 44, no. 5, pp. 831–848, 2006, doi: 10.1080/00207540500272519.

[35] L. D. Dye and J. S. Pennypacker, "Project portfolio management and managing multiple projects: two sides of the same coin?," in *Project Management Institute Annual Seminars & Symposium*, 2000.

[36] K. Yaghootkar and N. Gil, "The effects of schedule-driven project management in multi-project environments," *International Journal of Project Management*, vol. 30, no. 1, pp. 127–140, 2012, doi: 10.1016/j.ijproman.2011.02.005.

[37] C. Y. Chen, H. A. Liu, and J. Y. Song, “Integrated projects planning in IS departments: A multi-period multi-project selection and assignment approach with a computerized implementation,” *European Journal of Operational Research*, vol. 229, no. 3, pp. 683–694, 2013, doi: 10.1016/j.ejor.2013.02.057.

[38] T. Karrbom Gustavsson, “Organizing to avoid project overload: The use and risks of narrowing strategies in multi-project practice,” *International Journal of Project Management*, vol. 34, no. 1, pp. 94–101, 2016, doi: 10.1016/j.ijproman.2015.10.002.

[39] H. T. Tsai, H. Moskowitz, and L. H. Lee, “Human resource selection for software development projects using Taguchi’s parameter design,” *European Journal of Operational Research*, vol. 151, no. 1, pp. 167–180, 2003, doi: 10.1016/S0377-2217(02)00600-8.

[40] M. A. Kaulio, “Project leadership in multi-project settings: Findings from a critical incident study,” *International Journal of Project Management*, vol. 26, no. 4, pp. 338–347, 2008, doi: 10.1016/j.ijproman.2007.06.005.

[41] S. Elonen and K. A. Artto, “Problems in managing internal development projects in multi-project environments,” *International Journal of Project Management*, vol. 21, no. 6, pp. 395–402, 2003, doi: 10.1016/S0263-7863(02)00097-2.

[42] A. Belout and C. Gauvreau, “Factors influencing project success: The impact of human resource management,” *International Journal of Project Management*, vol. 22, no. 1, pp. 1–11, 2004, doi: 10.1016/S0263-7863(03)00003-6.

[43] A. Barreto, M. de O. Barros, and C. M. L. Werner, “Staffing a software project: A constraint satisfaction and optimization-based approach,” *Computers and Operations Research*, vol. 35, no. 10, pp. 3073–3089, 2008, doi: 10.1016/j.cor.2007.01.010.

[44] L. Silva and A. P. Costa, “Decision model for allocating human resources in information system projects,” *International Journal of Project Management*, vol. 31, no. 1, pp. 100–108, 2013, doi: 10.1016/j.ijproman.2012.06.008.

[45] Y. P. Huo and J. Kearns, “Optimizing the Job-Person Match with Computerized Human Resource Information Systems,” *Personnel Review*, vol. 21, no. 2, pp. 3–18, 1992, doi: 10.1108/00483489210012017.

[46] M. Engwall and A. Jerbrant, “The resource allocation syndrome: The prime challenge of multi-project management?,” *International Journal of Project Management*, vol. 21, no. 6, pp. 403–409, 2003, doi: 10.1016/S0263-7863(02)00113-8.

[47] L. Bernstein and C. M. Yuhas, *Trustworthy Systems Through Quantitative Software Engineering*, 1st ed. Hoboken, NJ, US: John Wiley & Sons, 2005.

[48] L. D. Otero, G. Centeno, A. J. Ruiz-Torres, and C. E. Otero, “A systematic approach for resource allocation in software projects,” *Computers and Industrial Engineering*, vol. 56, no. 4, pp. 1333–1339, 2009, doi: 10.1016/j.cie.2008.08.002.

[49] A. B. Raiden, A. R. J. Dainty, and R. H. Neale, “Current barriers and possible solutions to effective project team formation and deployment within a large construction organisation,” *International Journal of Project Management*, vol. 22, no. 4, pp. 309–316, 2004, doi: 10.1016/j.ijproman.2003.08.002.

[50] R. Chen, C. Liang, D. Gu, and J. Y. T. Leung, “A multi-objective model for multi-project scheduling and multi-skilled staff assignment for IT product development considering competency evolution,” *International Journal of Production Research*, vol. 55, no. 21, pp. 6207–6234, 2017, doi: 10.1080/00207543.2017.1326641.

[51] C. Ebert and P. De Neve, “Surviving Global Software Development,” *IEEE Software*, vol. 18, no. 2, pp. 62–69, 2001, doi: 10.1109/52.914748.

[52] Project Management Institute, *The Agile practice guide*, 1st ed. Newtown Square, PA, US: Project Management Institute, 2017.

[53] S. T. March and T. J. Vogus, “Design Science in the Management Disciplines,” in *Design Research in Information Systems*, 1st ed., A. Hevner and S. Chatterjee, Eds. Boston, MA, US: Springer, 2010, pp. 195–208.

[54] A. M. Beckers and M. Z. Bsat, “A DSS Classification Model for Research in Human Resource Information Systems,” *Information Systems Management*, vol. 19, no. 3, pp. 41–50, 2002, doi: 10.1201/1078/43201.19.3.20020601/37169.6.

[55] R. Sharda, D. Delen, and E. Turban, *Business Intelligence and Analytics: Systems for Decision Support*, 10th ed. Upper Saddle River, NJ, US: Pearson Education, 2015.

[56] D. B. Meinert and D. L. Davis, “Human Resource Decision Support Systems (HRDSS): Integrating decision support and human resource information systems,” *Information Resources Management Journal*, vol. 2, no. 1, pp. 41–48, 1989, doi: 10.4018/irmj.1989010104.

[57] H. Jantan, A. R. Hamdan, and Z. A. Othman, “Intelligent Techniques for Decision Support System in Human Resource Management,” in *Decision Support Systems, Advances in*, 1st ed., G. Devlin, Ed. InTech, 2010, pp. 261–276.

[58] J. P. Shim, M. Warkentin, J. F. Courtney, D. J. Power, R. Sharda, and C. Carlsson, “Past, present, and future of decision support technology,” *Decision Support Systems*, vol. 33, no. 2, pp. 111–126, 2002, doi: 10.1016/S0167-9236(01)00139-7.

[59] A.-K. Masum, L.-S. Beh, A.-K. Azad, and K. Hoque, “Intelligent Human Resource Information System (i-HRIS): A Holistic Decision Support Framework for HR Excellence,” *The International Arab Journal of Information Technology*, vol. 15, no. 1, pp. 121–130, 2018.

[60] M. Di Penta, M. Harman, and G. Antoniol, “The use of search-based optimization techniques to schedule and staff software projects: An approach and an empirical study,” *Software - Practice and Experience*, vol. 41, no. 5, pp. 495–519, 2011, doi: 10.1002/spe.1001.

[61] T. S. Whitman and A. C. Hyde, “HRIS: Systematically Matching the Right Person to the Right Position,” *Defense Management Journal*, vol. 14, no. 2, pp. 29–34, 1978.

[62] G. DeSanctis, “Human Resource Information Systems: A Current Assessment,” *MIS Quarterly*, vol. 10, no. 1, pp. 15–26, 1986.

[63] F. Ferrucci, M. Harman, and F. Sarro, “Search-Based Software Project Management,” in *Software Project Management in a Changing World*, 1st ed., G. Ruhe and C. Wohlin, Eds. Heidelberg, Germany: Springer, 2014, pp. 373–399.

[64] C. Besner and B. Hobbs, “An Empirical Identification of Project Management Toolsets and a Comparison among Project Types,” *Project Management Journal*, vol. 43, no. 5, pp. 24–46, 2012, doi: 10.1002/pmj.21292.

[65] Microsoft, “Project Management Software - Microsoft Project,” 2020. https://www.microsoft.com/en-us/microsoft-365/project/project-management-software (accessed Jun. 18, 2020).

[66] Silverbucket, “Resource Planning Tool for Project Business - Silverbucket,” 2020. https://www.silverbucket.com/ (accessed Jun. 18, 2020).

[67] Hub Planner, “Resource Management, Scheduling, Timesheets, Planning - Hub Planner,” 2020. https://hubplanner.com/ (accessed Jun. 18, 2020).

[68] Mavenlink, “Project Management Software - Mavenlink,” 2020. https://www.mavenlink.com/ (accessed Jun. 18, 2020).

[69] Smartsheet, “10,000ft by Smartsheet,” 2020. https://www.smartsheet.com/platform/10000ft (accessed Jun. 18, 2020).

[70] Zoho, “Zoho People,” 2020. https://www.zoho.com/people/ (accessed Jun. 18, 2020).

[71] Teamdeck, “Teamdeck - Complete resource management software for teams,” 2020. https://teamdeck.io/ (accessed Jun. 18, 2020).

[72] Saviom, “Enterprise Resource Management Software - Saviom,” 2020. https://www.saviom.com/resource-management-software.php (accessed Jun. 18, 2020).

[73] Clarizen, “Clarizen One - Manage work, automate workflows, collaborate like never before,” 2020. https://www.clarizen.com/product/clarizen-one/ (accessed Jun. 18, 2020).

[74] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information Systems Research,” *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, doi: 10.2307/25148625.

[75] K. Peffers, T. Tuunanen, and B. Niehaves, “Design science research genres: introduction to the special issue on exemplars and criteria for applicable design science research,” *European Journal of Information Systems*, vol. 27, no. 2, pp. 129–139, 2018, doi: 10.1080/0960085X.2018.1458066.

[76] R. Baskerville, “What design science is not,” *European Journal of Information Systems*, vol. 17, no. 5, pp. 441–443, 2008, doi: 10.1057/ejis.2008.45.

[77] S. Gregor and A. R. Hevner, “Positioning and Presenting Design Science Research for Maximum Impact,” *MIS Quarterly*, vol. 37, no. 2, pp. 337–355, 2013.

[78] V. Vaishnavi, B. Kuechler, and S. Petter, “Design Science Research in Information Systems,” *DESRIST*, 2019. http://www.desrist.org/design-research-in-information-systems/ (accessed May 12, 2020).

[79] P. Johannesson and E. Perjons, *An introduction to design science*, 1st ed. Heidelberg, Germany: Springer, 2014.

[80] S. T. March and G. F. Smith, “Design and natural science research on information technology,” *Decision Support Systems*, vol. 15, no. 4, pp. 251–266, 1995, doi: 10.1016/0167-9236(94)00041-2.

[81] R. Winter, “Design science research in Europe,” *European Journal of Information Systems*, vol. 17, no. 5. Palgrave Macmillan Ltd., pp. 470–475, 2008, doi: 10.1057/ejis.2008.44.

[82] E. Engström, M.-A. Storey, P. Runeson, M. Höst, and M. T. Baldassarre, “How software engineering research aligns with design science: A review,” 2019.

[83] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007, doi: 10.2753/MIS0742-1222240302.

[84] R. J. Wieringa, *Design science methodology for information systems and software engineering*, 1st ed. Heidelberb, Germany: Springer, 2014.

[85] Ł. Ostrowski and M. Helfert, “Commonality in various design science methodologies,” in *Proceedings of the Federated Conference on Computer Science and Information Systems*, 1st ed., M. Ganzha, L. Maciaszek, and M. Paprzycki, Eds. Los Alamitos, CA, US: IEEE Computer Society Press, 2011, pp. 317–320.

[86] J. Iivari, “Twelve Theses on Design Science Research in Information Systems,” in *Design Research in Information Systems*, 1st ed., A. Hevner and S. Chatterjee, Eds. Boston, MA, US: Springer, 2010, pp. 43–62.

[87] S. Morana *et al.*, “Tool support for design science research—Towards a software ecosystem: A report from a DESRIST 2017 workshop,” *Communications of the Association for Information Systems*, vol. 43, no. 1, pp. 237–256, 2018, doi: 10.17705/1CAIS.04317.

[88] J. Venable, J. Pries-Heje, and R. Baskerville, “FEDS: A Framework for Evaluation in Design Science Research,” *European Journal of Information Systems*, vol. 25, no. 1. Palgrave Macmillan Ltd., pp. 77–89, 2016, doi: 10.1057/ejis.2014.36.

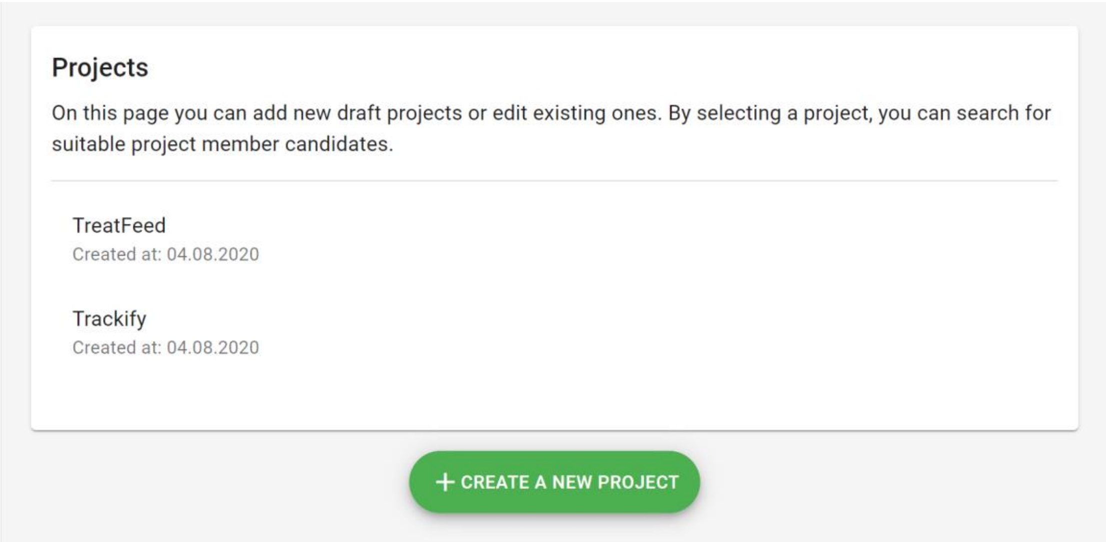
[89] B. Fortuna *et al.*, “QMiner: Data Analytics Platform for Processing Streams of Structured and Unstructured Data,” in *Software Engineering for Machine Learning Workshop, Neural Information Processing Systems*, 2014, doi: 10.13140/RG.2.1.4054.1283.

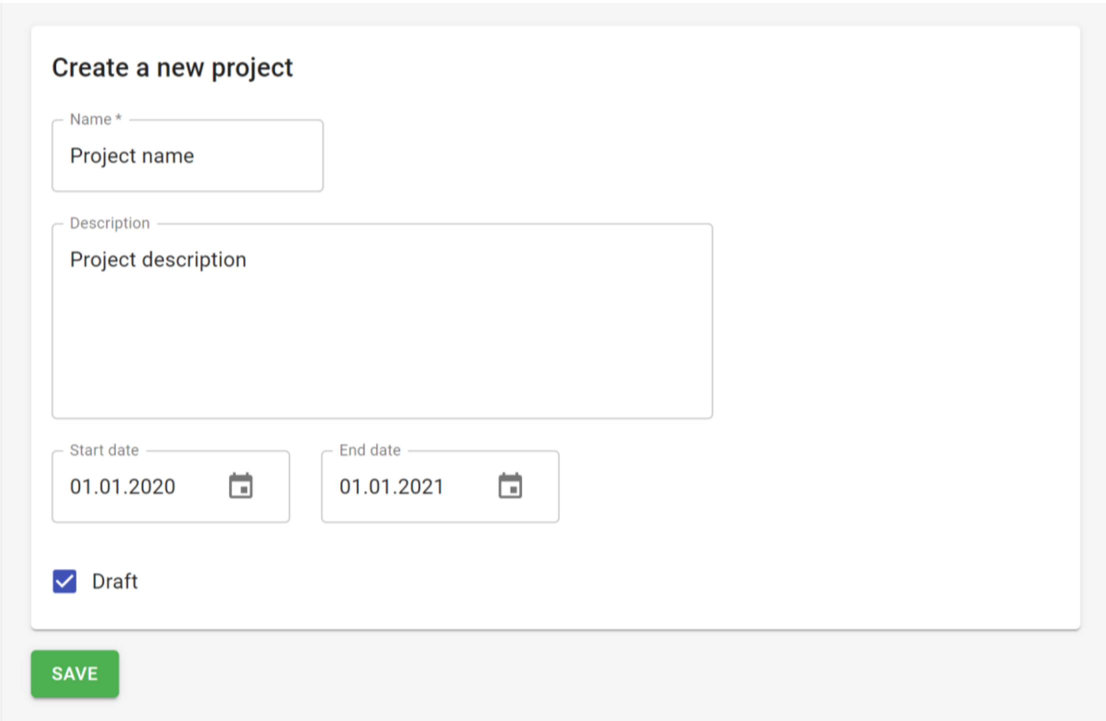
APPENDIX 1 Artefact wireframes

APPENDIX 2 Artefact screenshots

This appendix includes screenshots from the artefact. A 2.1 contains screenshots related to projects, such as project and project position creation, and employee search. 0 contains screenshots related to employees, such as employee and employee experience creation.

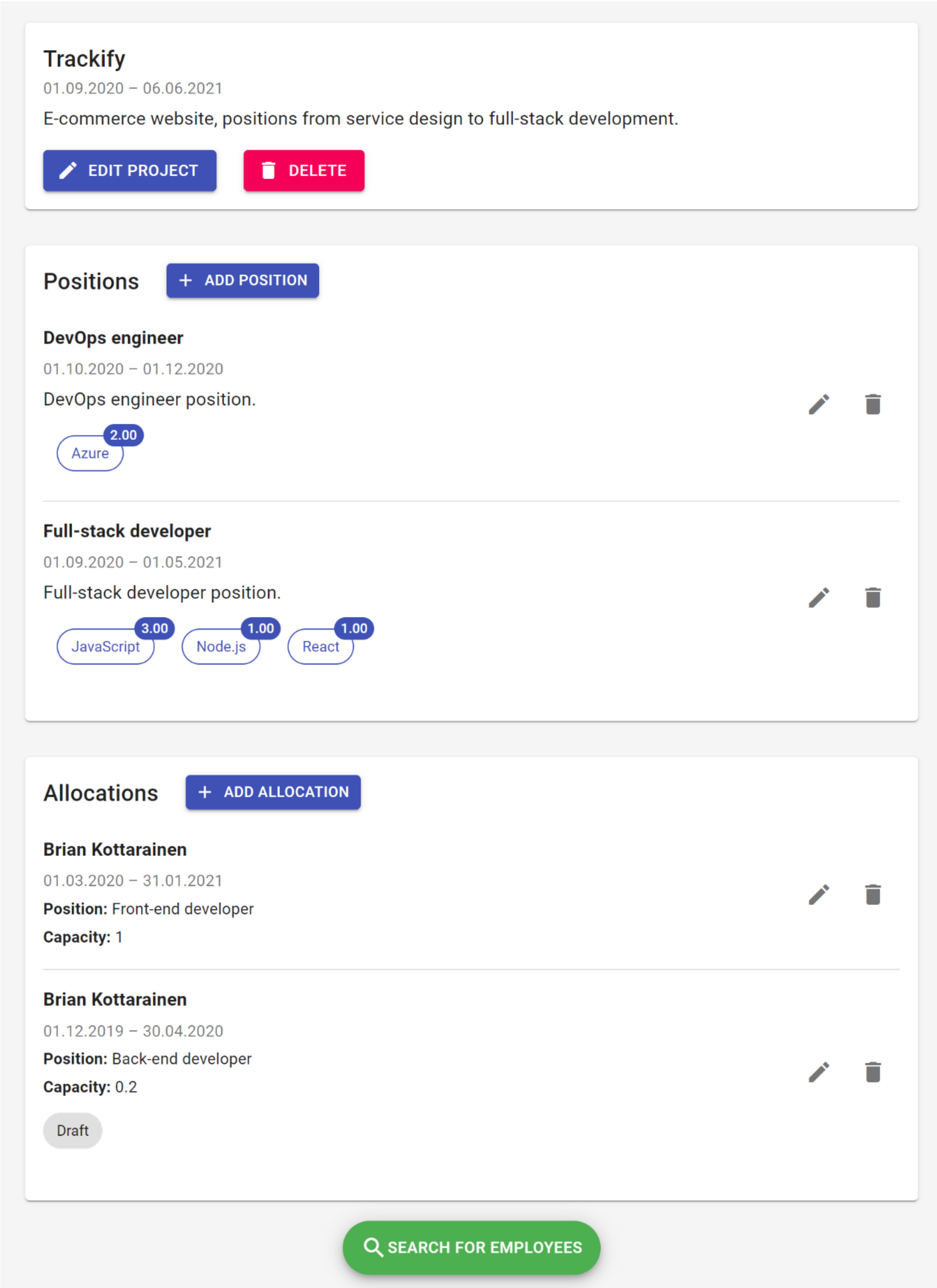
A 2.1 Projects





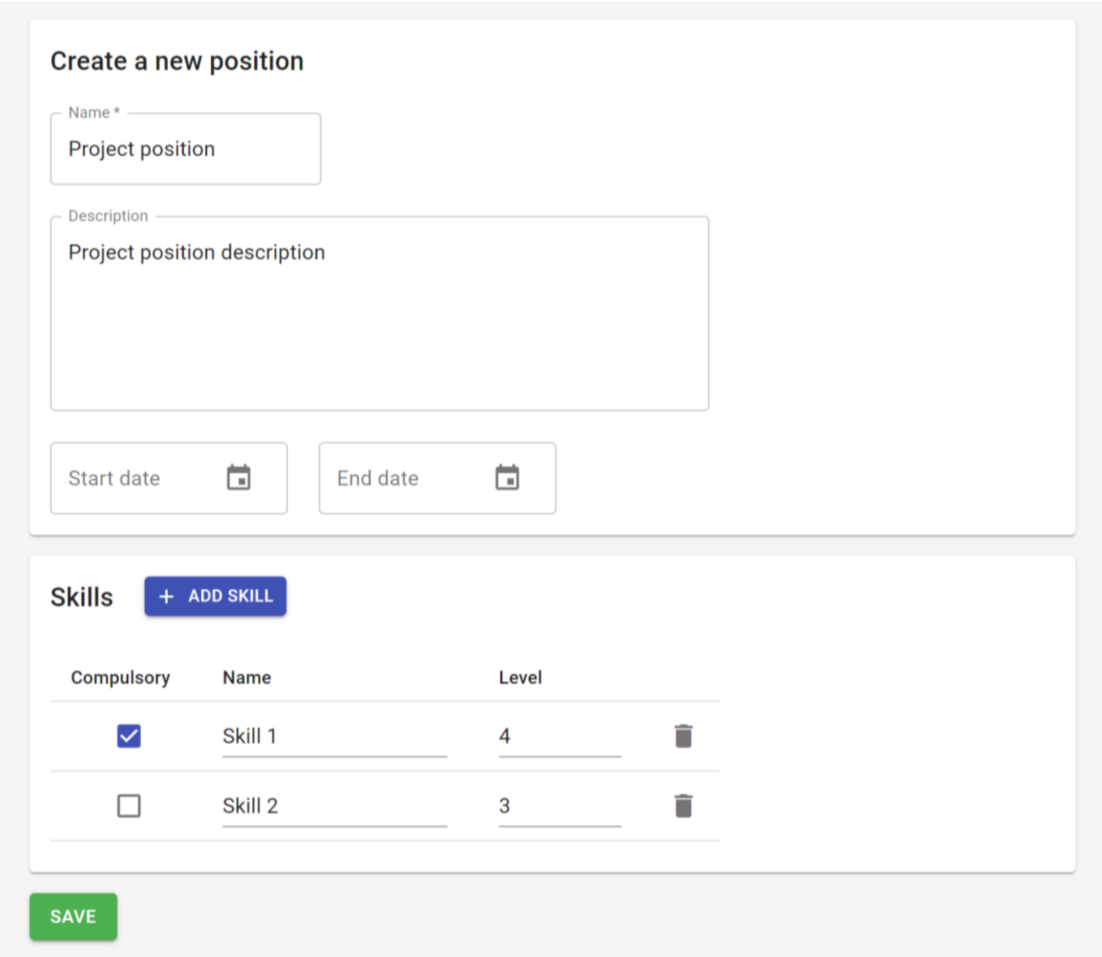
(continues)

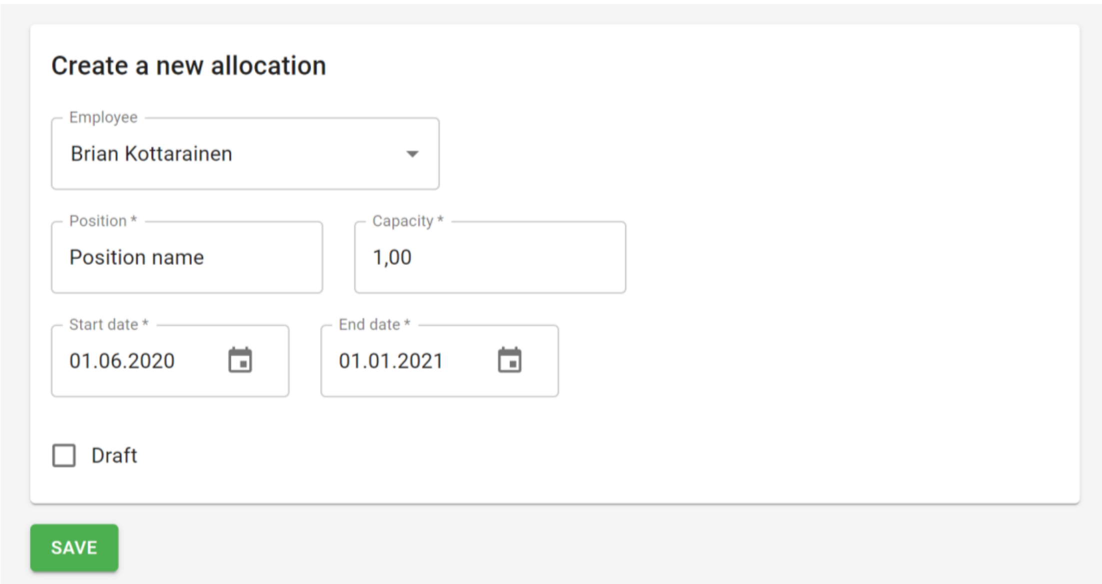
APPENDIX 2 (continues)



(continues)

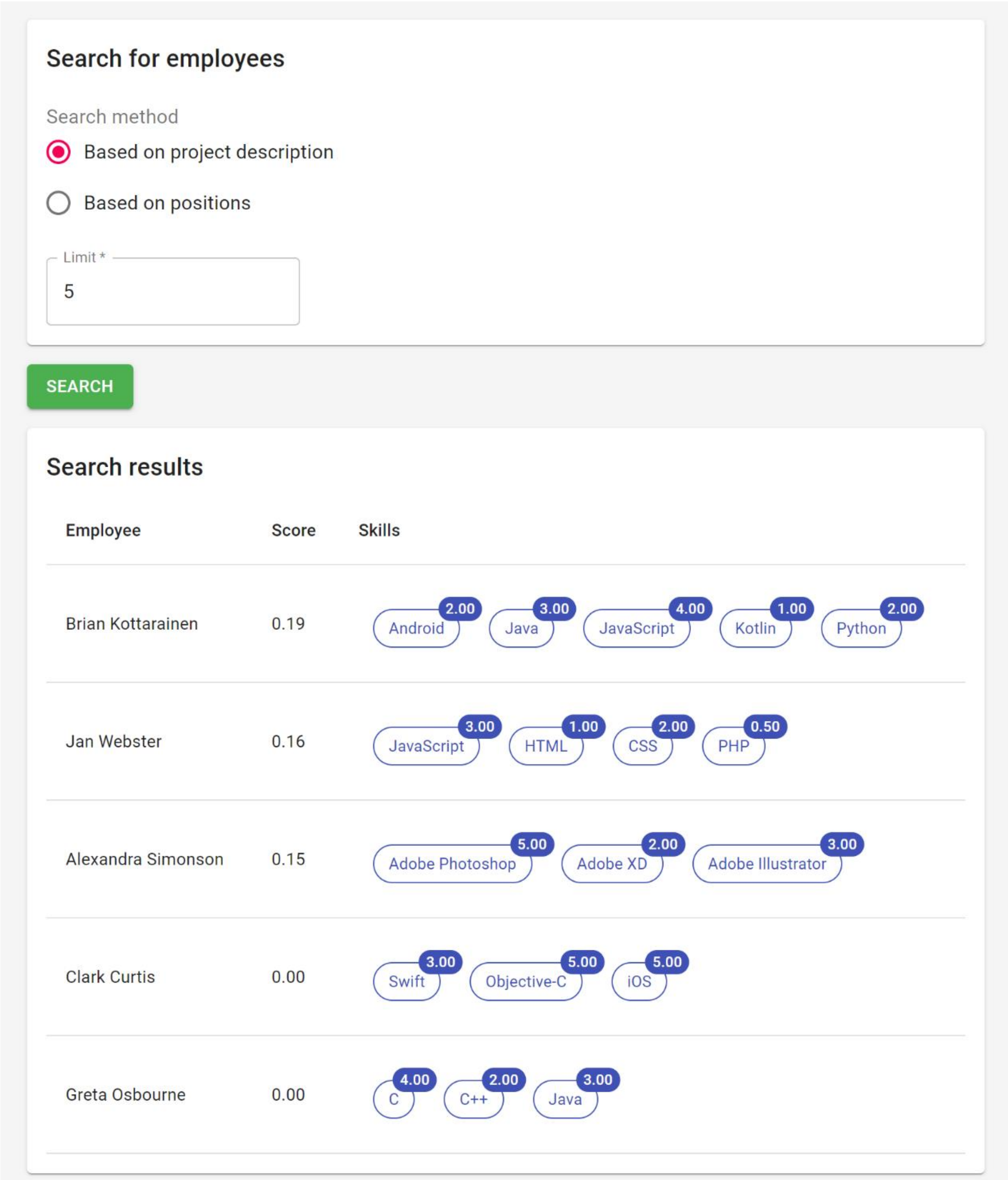
APPENDIX 2 (continues)





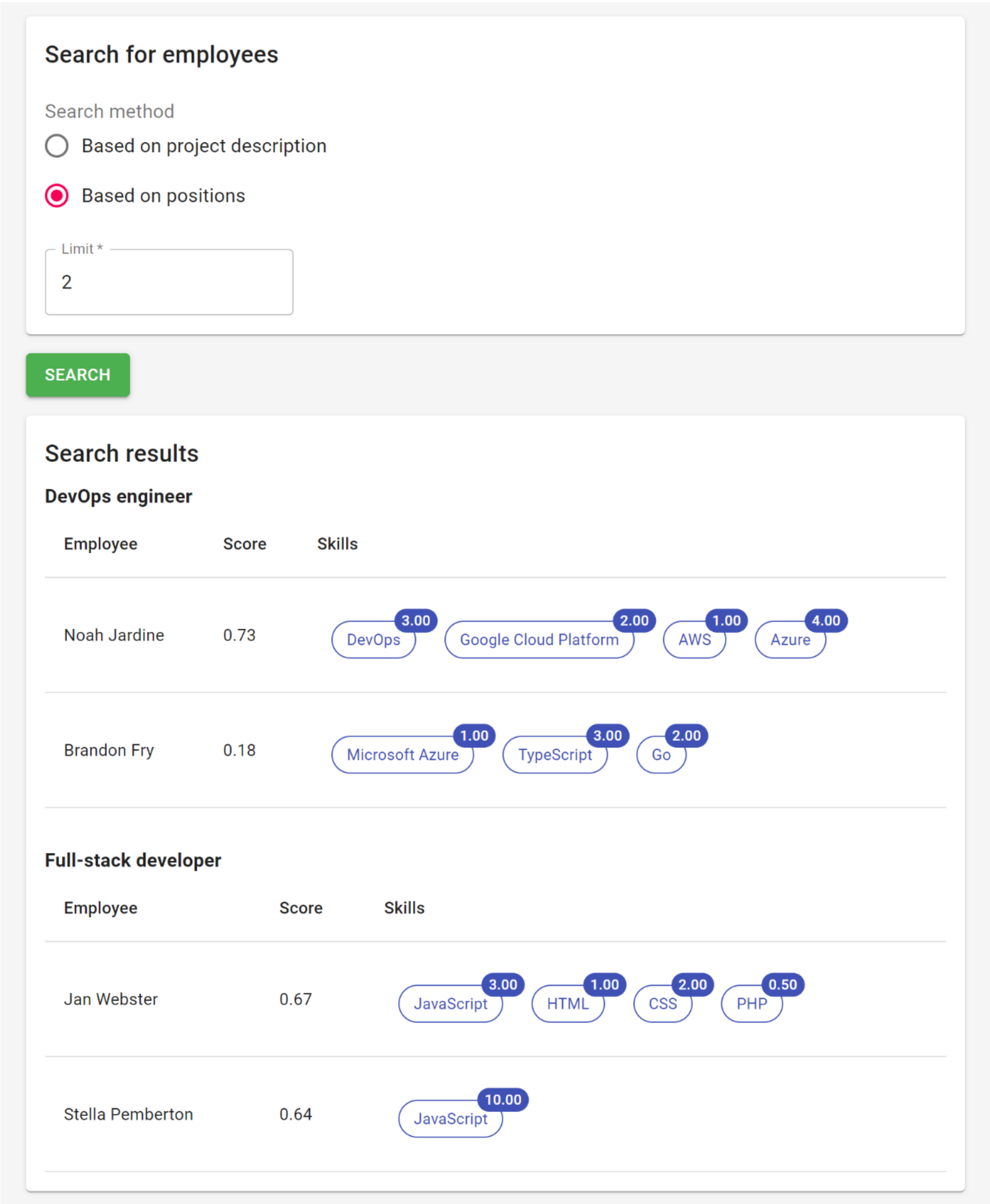
(continues)

APPENDIX 2 (continues)



(continues)

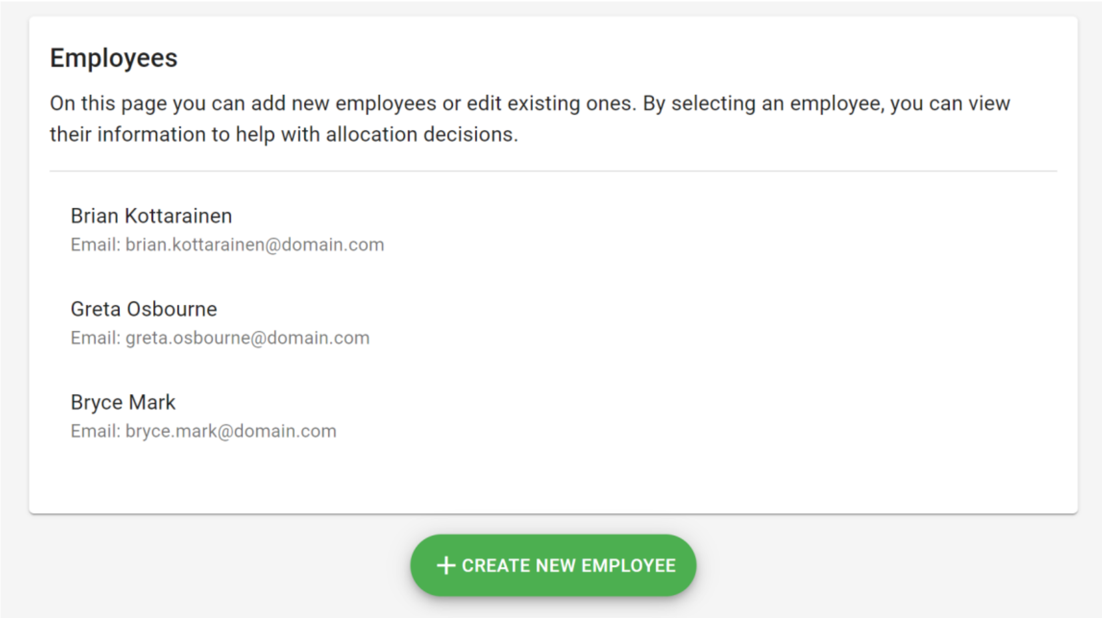
APPENDIX 2 (continues)

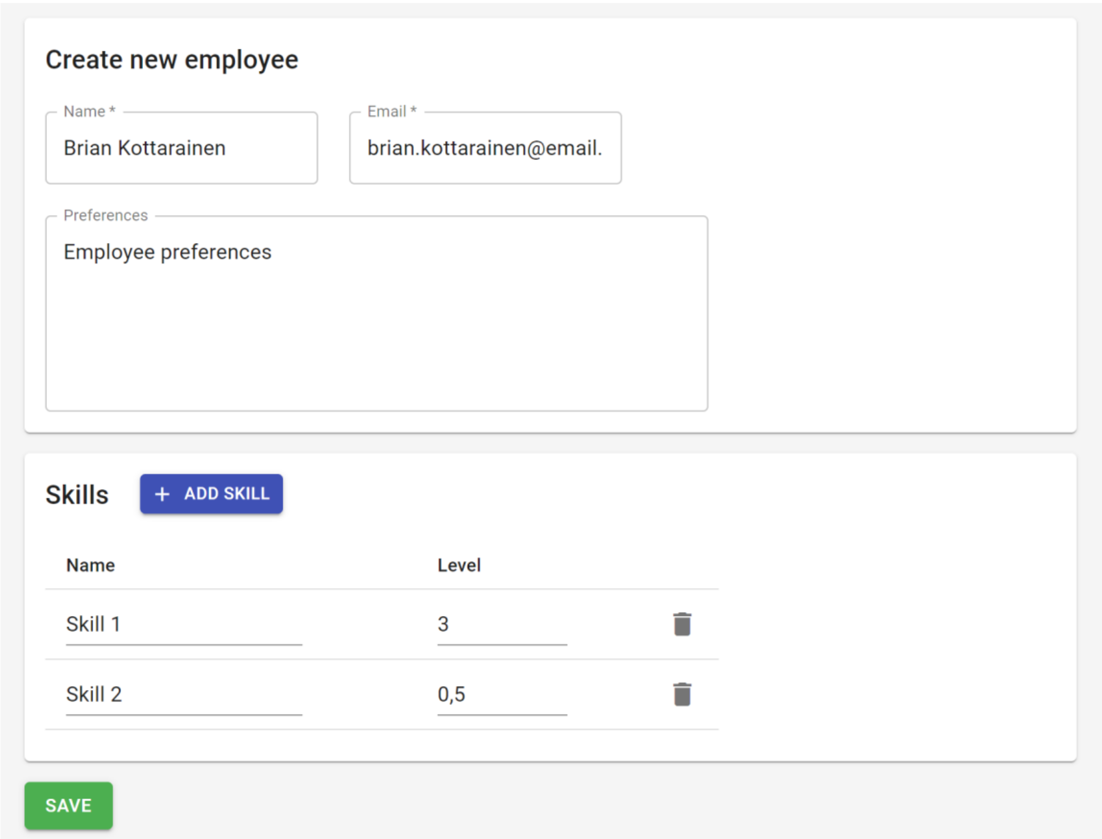


(continues)

APPENDIX 2 (continues)

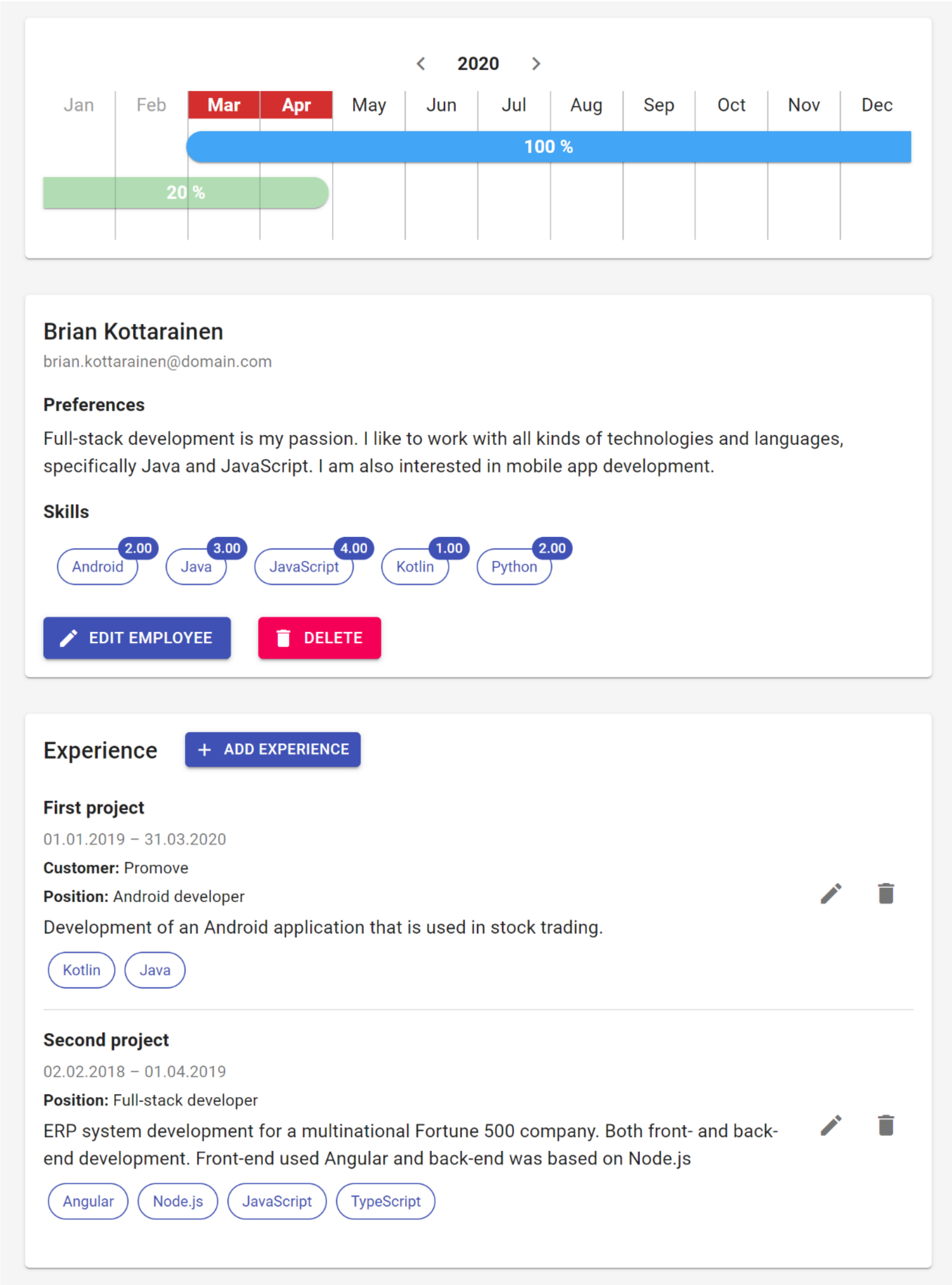
A 2.2 Employees





(continues)

APPENDIX 2 (continues)



(continues)

APPENDIX 2 (continues)