

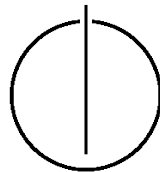
FAKULTÄT FÜR INFORMATIK

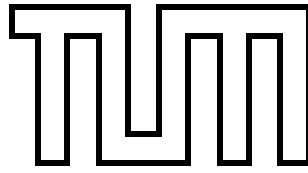
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

**Design and Implementation of a Mobile
Application for the Collaborative Structuring
of Knowledge-Intensive Processes**

Natascha Abrek





FAKULTÄT FÜR INFORMATIK

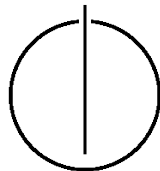
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Design and Implementation of a Mobile Application for
the Collaborative Structuring of Knowledge-Intensive
Processes

Design und Implementierung einer Mobilen
Anwendung zur Kollaborativen Strukturierung
Wissensintensiver Prozesse

Author:	Natascha Abrek
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Matheus Hauder, M.Sc.
Submission date:	October 14, 2015



I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, October 14, 2015

Natascha Abrek

Acknowledgements

I want to express my deepest gratitude to all those people who supported me throughout the course of my thesis. I want to thank Prof. Dr. Florian Matthes for the opportunity to work on such an interesting project. I especially want to thank my advisor Matheus Hauder for his constant support, guidance and advice during this whole time.

Special thanks goes to my dear husband Adam who is the greatest supporter of everything I do, for encouraging and motivating me, for always being understanding and helpful through busy times.

Abstract

The expanding role of knowledge in today's information-based economy has led businesses to increase their focus on process related work. Modelling processes of knowledge artifacts within so called knowledge-intensive processes (KIPs) became an essential task for organizations. KIPs comprise activities such as building, sharing and reuse of knowledge as well as the collaboration and cooperation of workers. However, KIPs take hold of an unpredictable and dynamic nature and are characterized by high variability and changeability. These factors establish the need to support such processes with appropriate methods. To facilitate the structuring of knowledge-intensive processes a large variety of supportive tools has emerged in recent years. Especially with advances in mobile technology the necessity to support knowledge work through mobile devices has increased. Businesses are increasingly incorporating mobile technologies for communication and collaboration purposes. Nonetheless, mobile applications need to meet different usability requirements than web applications. Usage contexts and interaction methods differ from those of desktop websites. To provide a successful user experience several key aspects such as available mobile technologies, compatibility and the selection of essential information to be displayed need to be further investigated.

In this work, a mobile web application for collaborative structuring of knowledge-intensive processes is designed and implemented. Thereby, a mobile tailored version of the existing web application Darwin is developed. Best practices and usability requirements towards mobile applications are gathered and integrated. Furthermore, an evaluation of the developed mobile solution regarding usability aspects such as perceived user satisfaction, efficiency and ease of use is conducted. Revised design solutions based on the evaluation results are then incorporated into the iterative design process.

Zusammenfassung

Die zunehmend wichtige Rolle von Wissen in der heutigen Informationsgesellschaft hat dazu geführt, dass sich Unternehmen verstärkt auf prozessbezogene Aufgaben konzentrieren. Das Modellieren von Artefakten sogenannter wissensintensiver Prozesse ist zu einer grundlegenden Aufgabe für Organisationen geworden. Zu den Aktivitäten wissensintensiver Prozesse gehören das Generieren, Teilen und Nutzen von Wissen sowie die Kollaboration und Kooperation zwischen den Wissensarbeitern. Derartige Prozesse zeichnen sich jedoch durch ihre Unvorhersehbarkeit und Dynamik sowie eine hohe Variabilität und Wechselhaftigkeit aus. Aufgrund dieser Faktoren müssen angemessenen Methoden zur Unterstützung wissensintensiver Prozesse gefunden werden. Eine Vielzahl von Hilfswerkzeugen wurde in den letzten Jahren entwickelt, um die kollaborativen Aspekte von wissensintensiven Prozessen zu unterstützen. Ebenso machen die Fortschritte in der mobilen Kommunikationstechnologie den Einsatz von mobilen Endgeräten innerhalb der Unternehmen zur Unterstützung von Wissensarbeitern zur Notwendigkeit. Unternehmen setzen verstärkt mobile Technologien für die Kommunikation und Kollaboration ein. Nichtsdestotrotz müssen mobile Anwendungen andere Kriterien erfüllen als Webanwendungen. Nutzungskontext und Interaktionsmethoden unterscheiden sich von üblichen Desktop-Websites. Um eine gute User Experience zu gewährleisten, müssen einige grundlegende Aspekte wie verfügbare mobile Technologien, Kompatibilität und die Auswahl relevanter Inhalte genauer untersucht werden.

In dieser Arbeit wird eine mobile Webanwendung für das kollaborative Strukturieren wissensintensiver Prozesse gestaltet und implementiert. Dazu wird eine auf Mobiltelefone adaptierte Version der existierenden Darwin Applikation entwickelt. Richtlinien und Anforderungen an die Nutzerfreundlichkeit von mobilen Anwendungen werden untersucht und integriert. Des Weiteren werden die Ergebnisse einer Nutzerbefragung vorgestellt, die Informationen bezüglich der Nutzerzufriedenheit, Effizienz und Bedienbarkeit der entwickelten Lösung preisgeben soll. Basierend auf den Ergebnissen der Nutzerbefragung werden Gestaltungslösungen verbessert und in einem iterativen Prozess in die Anwendung eingebunden.

Contents

Acknowledgements	vii
Abstract	ix
I. Overview and Background	1
1. Motivation	3
1.1. Problem Description	4
1.2. Research Scope	5
1.3. Outline	6
2. Process Support for Knowledge-Intensive Work	9
2.1. Terms and Definitions	9
2.1.1. Knowledge Work	10
2.1.2. Knowledge Worker	10
2.1.3. Knowledge-Intensive Processes	11
2.2. Characteristics of Knowledge-Intensive Processes	12
2.3. Requirements for Knowledge-Intensive Processes	14
3. Mobile Usability and User Experience	17
3.1. Definitions	17
3.1.1. Usability	17
3.1.2. User Experience	19
3.2. Usability attributes	20
3.3. Usability guidelines	20
3.3.1. Eight Golden Rules by Shneiderman	21
3.3.2. Nielsen's Usability Heuristics	22
3.3.3. Usability Standards	23
3.4. Designing for Mobile Interfaces	24
3.4.1. Mobile Websites vs. Desktop Websites	24
3.4.2. Interface Guidelines for Mobile Devices	25
4. Related Work	27
4.1. Mobile Support for Knowledge-Intensive Processes	27
4.2. Task-Centered Mobile Applications	30

II. Conceptual Design	35
5. A User-Centered Design Approach for Mobile Websites	37
5.1. User-Centered Design	37
5.1.1. Key Principles	37
5.1.2. Design Process for Darwin	38
5.2. User Analysis	39
5.2.1. User Groups	39
5.2.2. User Environment	40
5.3. Requirements Specification	41
5.3.1. Organizing Content	41
5.3.2. Structuring KIPs with Tasks	44
5.3.3. Enhancing Visibility of Work Activities	45
5.3.4. Managing Users	46
5.3.5. Design Requirements	47
5.4. Design Solutions	49
5.5. Assessment of Mobile Technologies	49
5.5.1. Mobile Website vs. Native Application	49
5.5.2. Mobile Frameworks	50
III. Implementation and Evaluation	53
6. Technical Implementation	55
6.1. System Architecture	55
6.1.1. Client-Side Architecture	56
6.1.2. Server-Side Architecture	56
6.2. Technologies	57
6.3. Routing	57
6.4. Data Model	59
6.5. Darwin API	60
7. Design of the Mobile User Interface	61
7.1. Layout and Styling	61
7.1.1. Structure	61
7.1.2. Color and Typography	62
7.1.3. Theming	62
7.2. User Interface Components	63
7.2.1. Navigation	64
7.2.2. Activity Feed	65
7.2.3. Wiki Pages	68
7.2.4. Profile Page	72
7.2.5. Secondary Pages	73
8. User Evaluation	75
8.1. Usability Evaluation Methods	75

8.2. Empirical Evaluation of Darwin Mobile	76
8.2.1. Test Objectives	77
8.2.2. Methodology	77
8.2.3. Tasks	78
8.2.4. Participants	79
8.2.5. Procedure	79
8.2.6. Results and Analysis	80
8.2.7. Discussion	87
 IV. Conclusion	 91
 9. Conclusion and Outlook	 93
9.1. Summary	93
9.2. Outlook	94
 Appendix	 99
 Bibliography	 109
 List of Figures	 113
 List of Tables	 116

Part I.

Overview and Background

1. Motivation

The significant rise of knowledge-centric organizations has become apparent in today's global economy. In recognition of the potential and influence knowledge can have on businesses, a shift from industrial and operational based work to knowledge-intensive work has taken place [5, 49]. In the manual work process materials are manufactured into a visual product. In modern organizations, knowledge has become the new means of production with a different product outcome. As the new driver of economic growth knowledge is produced, processed and distributed by experts. This transition to an information society leads to universal changes in the competitive environment of almost all industrial sectors. Knowledge has become a high-value investment in the information-based economy and managing knowledge-intensive processes (KIPs) an essential task to economic performance.

However, managing rising volumes of information and knowledge requires different development approaches and management structures. While it is easier to control and supervise manual work, which uses physical assets such as work equipment or machines, knowledge-based work has an intangible character making activities of knowledge workers less visible and more challenging to supervise [12]. Workers generate, organize and use their knowledge in the production process giving them the ownership and control over the main asset of production. Moreover, knowledge is widely distributed over organizations. Knowledge workers need to collaborate and share their knowledge through multiple levels of management in an organization. These new conditions force organizations to overthink and restructure their strategies. Drucker, one of the first people to define the concept of knowledge work, stated that increasing the productivity of knowledge workers will become the main contribution of an organization's management [15]. Former means of control such as strict work plans are not enough to effectively manage KIPs. Optimization strategies are necessary to better coordinate human activities and interactions within knowledge-intensive processes. These processes need to become more transparent, efficient and easier to manage. Thus, many organizations try to utilize the power of advanced information technology to improve existing processes.

Just as the workforce of knowledge workers has grown, the focus on KIPs and ways to manage them has fortified. Information systems aims to support KIPs and their evolution during the process execution. They expand the possibilities to communicate and share knowledge among employees. Physical barriers are eliminated, improving collaboration and communication between geographically separated workers. However, knowledge-intensive processes take hold of an unpredictable, emergent and uncertain manner [12]. Existing business information systems do not provide the necessary means to support KIPs. The reuse of formerly structured processes is not always feasible. Changes in the execution environment or infrastructure such as different business constraints or security policies may lead to necessary adaptations of the process. Furthermore, KIPs make immense

use of collaborative features which are hardly controllable. Hence, the use of convenient approaches to manage such processes is immensely important to ensure efficiency and effectiveness of an organization.

Furthermore, as mobility has become an important factor in today's economy, KIP management approaches to support mobile knowledge work are particularly indispensable. Mobile devices have established themselves as new computing platforms. Internet-based businesses such as Twitter or Google illustrate the importance of a mobile strategy. Not only for consumers, but also for businesses mobile access to data can serve as a great benefit. Technological improvements have also surged an increasing demand of mobile usage within organizations. More and more, the incorporation of mobile technology into different areas of the enterprise is reinforced. Cross-device applications make it easier than ever for workers to collaborate and communicate information. But other than in personal context, the business use holds more complex information and processes. Elaborated methods for mobile KIP support are still missing.

1.1. Problem Description

Today, many support systems, either for business processes, knowledge work or collaboration, are available as web-based applications. Unlike traditional software systems, they do not need to be installed on every standalone client machine. Advantages regarding portability, maintainability and ease of use make businesses move from traditional software systems to web applications. A major benefit of web applications is the platform independence. They provide easy access from all kinds of devices, including mobile phones. Compared to native mobile applications, mobile web applications are browser-based and can easily be accessed over the internet by a broad audience. However, not only a lack of mobile strategies to support businesses in their processes is apparent, mobile-optimized websites for businesses are scarce.

The first challenge that arises is the transfer of the complex desktop web application Darwin for KIP support to a mobile optimized application for the use on a lightweight, handheld device. Technical and hardware limitations such as smaller screen size, limited processing power and performance represent restrictions which need to be considered in the design and implementation process. The conditions and context in which a mobile web application is used also differ from their desktop counterparts. While the use of desktop computers aims at work in the office, mobile devices are directed for the usage outside the office or en route. Hence, an important task for the customization of a mobile system is the comprehension of suitable features which are relevant to the mobile use case. This can also result in restructuring or even reducing content for mobile browsers. While desktop pages can contain richer dialogs with high information density, in mobile interfaces page content and navigation options are limited.

The second challenge that arises is to provide support for knowledge-intensive processes by the mobile optimized website. While tools which aim to support KIPs for collaborative work start to appear on the market, KIPs clearly induce new challenges. The utilization of knowledge to perform tasks requires new, effective methods to support this kind of work

activity. This includes easy access to knowledge and an improvement of knowledge quality and distribution. Information and knowledge needs to be delivered in a well-defined context of tasks and deliverables at the right level of granularity. Many information systems aim to provide improvements to those aspects. They combine applications of storage technologies and groupware. To perform knowledge work, however, they still lack in appropriate methods due to the different characteristics of KIPs. Furthermore, end-users are often faced with difficulties when modeling processes. Especially mobile applications still fall behind when it comes to the management of KIPs. Hence, knowledge workers cannot yet utilize the full potential of the mobile technology to improve their processes.

The third challenge is to develop a user-friendly mobile interface. Good mobile usability contributes to the user's overall satisfaction with the system and helps him to achieve his work goals efficiently. Different conditions in the mobile environment lead to different factors and considerations when it comes to usability. Mobile devices for example employ new types of interaction patterns such as touch gestures. These new conditions surrounding the mobile usage need to be identified, analyzed and integrated into the system. This work attempts to overcome the challenges of mobile usability and aims to fulfill well-established quality attributes like intuitive design, efficiency of use or ease of learning in its presented solution.

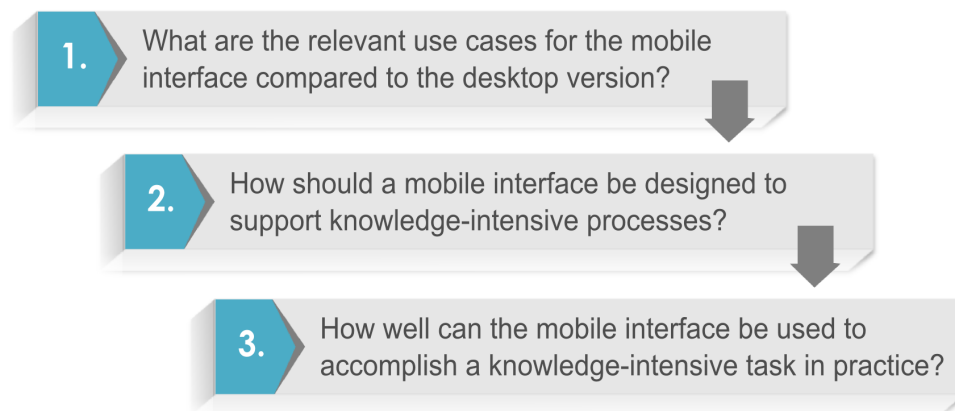


Figure 1.1.: Research questions.

1.2. Research Scope

This work focuses on three research questions which are pictured in Figure 1.1. They have been derived from the main challenges presented in the problem description of the previous section. The main objective of this work is to answer these research questions in order to develop a user-friendly, task-oriented mobile website to support collaborative knowledge work. Thereby, a user-centered design approach is applied to focus on user goals and

needs. To develop a mobile support system for knowledge workers their main tasks need to be identified and integrated as features into the mobile system. These features must satisfy the user expectations to collaboratively manage and structure knowledge-intensive processes in a mobile environment. During the development, design principles for mobile usability and user experience are considered. Specific requirements towards mobile devices are derived and implemented into the solution. As part of an iterative design process, a user evaluation is conducted to determine the perceived user experience and user satisfaction. The results of the user evaluation are analyzed and necessary changes are applied to create a revised solution according to the findings.

1.3. Outline

This thesis is divided into overall four parts. Parts I and IV cover the theoretical work done for this thesis. Part II and III comprise the practical work. Figure 1.2 gives an overview of the thesis outline.

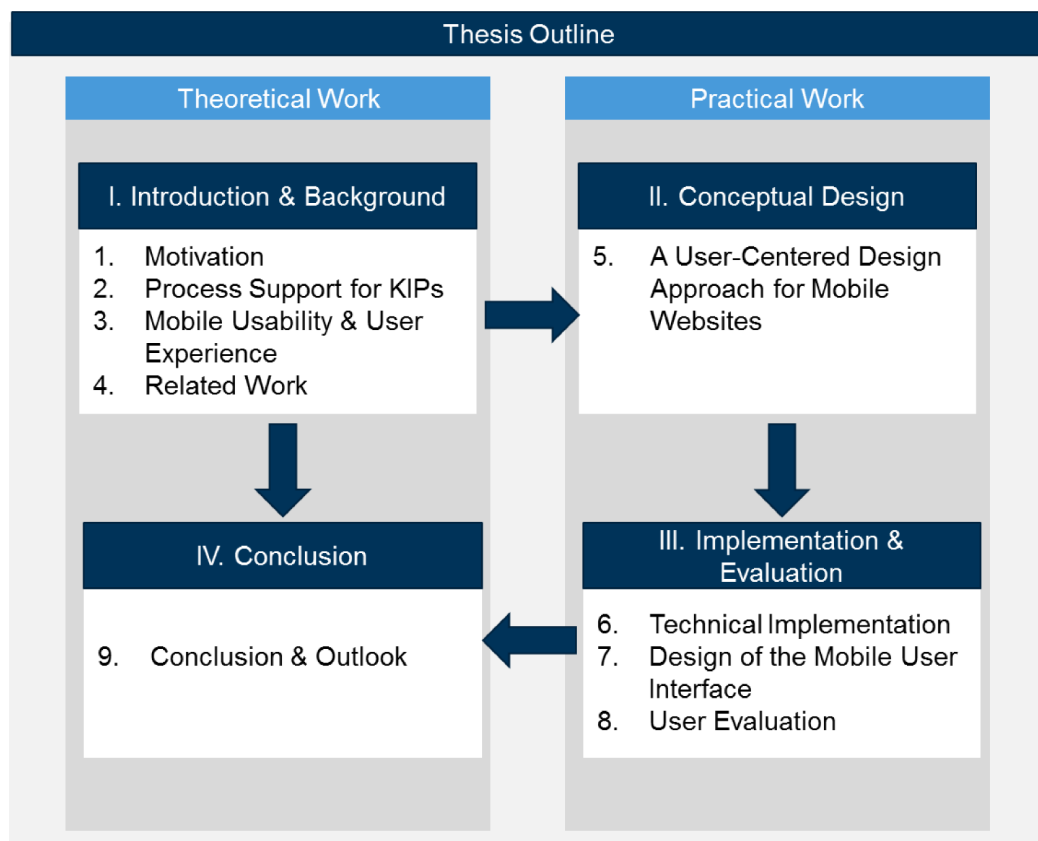


Figure 1.2.: Outline of the thesis.

The first part gives an introduction to the thesis as well as a summary of relevant background knowledge. Chapter 1 provides a brief overview of the overall thesis objective describing this work's motivational aspects and the thesis statement. It begins by stating the purpose of this work and the problem formulation. The scope of research defines the goal of the work and the approach to solve the identified problems. Chapter 2 introduces the fundamental background of knowledge-intensive processes. It covers the relevant terms and definition of knowledge work to comprehend the concept of knowledge-intensive processes. Characteristics of KIPs are presented and requirements for the management of these processes are analyzed. Chapter 3 presents the general concepts of usability and user experience. Design principles, usability attributes and norms for mobile user interfaces retrieved from literature are introduced. Finally, chapter 4 provides an overview of existing mobile applications and related work conducted in the past.

The second part introduces the conceptual design of the developed website implementing a user-centered design approach. In chapter 5, the general principles of a user-centered design process are explained. Then the phases of the applied process for this work are presented comprising a user analysis, requirements specification towards design and functionality and the developed design solutions. Finally, an assessment of the mobile technologies and frameworks is presented.

The third part covers the implementation and the evaluation of this work. Chapter 6 starts with the presentation of the technical aspects of the system including architecture design, used technologies, the data model and the API. Chapter 7 provides a detailed description of the developed mobile website. The overall structure and layout of the website as well as the applied mobile patterns are presented. Individual components and main features are illustrated with screenshots of the system. Chapter 8 comprises the conducted user evaluation. It holds a description of the evaluation design with methodology, participants and evaluation procedure. Finally, the results of the evaluation are discussed and the findings are presented.

The last part of this thesis is the conclusion. In the final chapter, the key points of this work are summarized, research contributions are highlighted and the overall implications of this work's findings are discussed. It furthermore gives a brief outlook to future work that can be done in this field of research.

2. Process Support for Knowledge-Intensive Work

To determine how knowledge workers can be supported in their processes it is necessary to grasp the differences between knowledge-intensive processes and non-knowledge-intensive processes. This chapter provides an overview of knowledge work and knowledge-intensive processes. Section 2.1 provides the definitions of knowledge work, knowledge worker and knowledge-intensive processes. Section 2.2 then describes the characteristics of knowledge-intensive processes. Finally, in section 2.3 the requirements for the support of KIPs are presented.

2.1. Terms and Definitions

Until the late 1920s, manual workers carried out two thirds of organizations, work outperforming knowledge workers [15]. By today, knowledge work has exceeded manual work. Knowledge work has an information-based nature. It is characterized as work requiring a formal education, expertise and high level of skills [39].

Characteristic	Knowledge Work	Manual Work
Input	Information/Knowledge	Material input
Output	Information/Knowledge	Tangible product or service
Work object	Intangible	Tangible
Activity	Processing of information, creating and using knowledge	Processing of material input
Skills	Cognitive skills, mental dexterity	Manual skills, manual dexterity
Tools	Information and communication	Tools for physical work

Table 2.1.: Differences between knowledge and manual work [48].

While results of manual work can be seen in form of converted materials during the production process, the output of knowledge work is mostly intangible. It consists of converting information from one form to another, generating and networking knowledge. Both input and output of knowledge work are primarily available as information, knowledge or data. Table 2.1 summarizes the different key characteristics between manual and knowledge work.

2.1.1. Knowledge Work

Knowledge work has been described in a number of studies leading to several views on this field of research. In his study about the production and distribution of knowledge in the United States economy, Machlup [41] defines knowledge work as the "production and transmission of knowledge". Stehr [61] extends this definition by classifying knowledge work as the "production and reproduction of information and knowledge". For this work, the definition of North and Gueldenberg is adopted:

Knowledge work is an activity based on cognitive skills that has an intangible result and whose value added relies on information processing and creativity, and consequently on the creation and communication of knowledge [48].

According to North and Gueldenberg, the value of knowledge work is the transformation of information to knowledge. Their definition further highlights the difference between knowledge and information. While information itself is inactive, the processing of information builds knowledge.

2.1.2. Knowledge Worker

The key to successfully support KIPs is to understand the capabilities, needs and goals of knowledge workers. They are responsible for exploring and realizing ideas and concepts. Knowledge workers can be found in a variety of fields, reaching from information technology to professions such as lawyers, physicians or teachers. Knowledge workers are highly independent and have a certain level of autonomy in their decision making process [73]. Definitions of knowledge workers are diverse in the literature. They examine knowledge workers from different views. Some definitions are based on the individual characteristics of knowledge workers, while others focus on work content or work output. Drucker defines knowledge workers based on their work content. According to Drucker, any person who works primarily with his knowledge is classified as a knowledge worker.

A knowledge worker is an employee whose major contribution depends on employing his knowledge rather than his muscle power and coordination, frequently contrasted with production workers who employ muscle power and coordination to operate machines [14].

2.1.3. Knowledge-Intensive Processes

In general, a process is often defined as a systematic sequence of actions or procedures leading to a particular result. A business process (BP) is the execution of tasks to achieve a defined business outcome [11]. Knowledge-intensive processes are tightly bound to human-centered knowledge. More precisely, a process is defined as knowledge-intensive when it relies heavily on the input of knowledge and knowledge activities. KIPs have an highly complex, unpredictable and dynamic character making it difficult to structure or automate them [12]. Unlike in business processes, activities of KIPs rarely repeat themselves. These characteristics can lead to a lack of transparency and traceability of work. Vaculin et al. [66] define KIPs as follows:

Knowledge-intensive processes are business processes whose conduct and execution are heavily dependent on knowledge workers performing various interconnected knowledge-intensive decision making tasks. KIPs are genuinely knowledge, information and data centric and require substantial flexibility at design and run time [66].

Mundbrod et al. [43] propose in their work a life cycle for collaborative knowledge-intensive business processes. The life cycle describes different stages of a KIP that need to be supported by an information system.

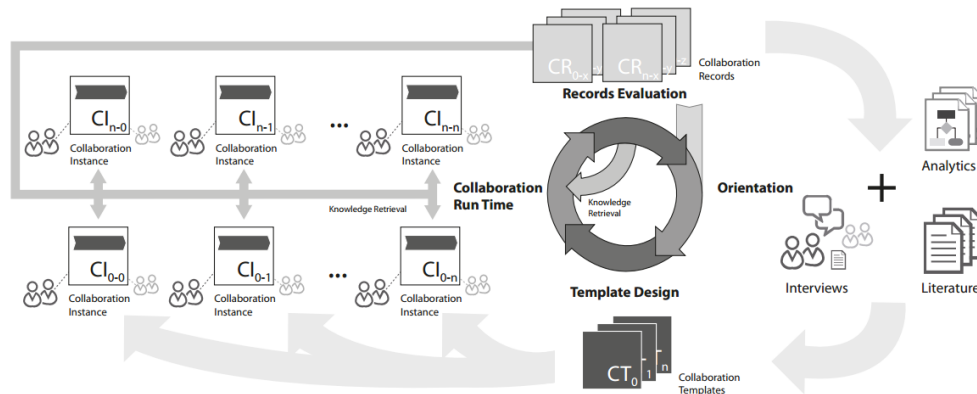


Figure 2.1.: The collaborative KIP life cycle [43].

The life cycle consists of four main phases. The first phase is the *orientation* phase. In this phase, all the information is gathered on how knowledge workers perform tasks collaboratively. This comprises different collaboration types, the communication and coordination structure and the information flows. Means to collect the information are interviews, literature reviews or analysis. The second phase is the *template design*. Thereby, goal-oriented collaboration templates are created. It contains all coordination artifacts and is employed by knowledge workers during run time. The third phase is the *collaboration run time*. In

this phase, workers instantiate a collaboration template to create a collaboration instance. A collaboration instance determines how to support the interaction between knowledge workers during run time. The final phase is the *records evaluation*. Knowledge workers can evaluate former collaboration records and compare them with new ones to gain insights about how to improve their collaborative templates. Figure 2.1 illustrates the complete life cycle of collaborative KIPs.

2.2. Characteristics of Knowledge-Intensive Processes

After establishing a basic understanding of the KIP life cycle, the characteristics of KIPs are now presented. While research has established a comprehensive understanding of business processes, knowledge-intensive processes are still being explored. To go beyond traditional BPs, the characteristics of KIPs need to be investigated. In the following, the key characteristics of KIPs derived from Di Ciccio et al.'s work [12] are presented. Their summary of KIP characteristics is based on a broad literature analysis.

Knowledge-driven: All KIPs are knowledge-driven. Thereby, it is differentiated between explicit and implicit, also called tacit, knowledge. Explicit knowledge is structured with a formal language in documents, artefacts and data. This kind of knowledge can be made verbal by the holder of the knowledge. Examples for explicit knowledge are best-practices, guidelines or instructions. Tacit knowledge, on the other hand, cannot be expressed verbally. It is action-oriented and practical, often acquired through personal experience [59]. Processes often depend on both types of knowledge. They can influence knowledge workers in their decisions. During a process instance, they can influence the flow of activities and events.

Collaboration-oriented: Knowledge is often differently distributed within businesses. Knowledge workers own different skills and levels of expertise. Hence, they need to continuously share their knowledge with other workers. The collaboration among knowledge workers adds high value to the workplace by processing, exchanging and creating new knowledge.

Unpredictable: Unpredictability is a characteristic which makes KIPs particularly difficult to manage. Process activities change based on the given context, environment and situation. Unlike in business processes, which elements can often be foreseen and planned, the course of actions in KIPs is more flexible.

Emergent: Activities of KIPs emerge during the execution of the process. The previous process progression determines the sequel of the process. Emergent actions result from situated decisions and available knowledge.

Goal-oriented: The evolution of a knowledge-intensive process goes along with a variety of intermediate objectives, decision points and milestones. Milestones can increase the competence of successfully managing processes. Setting deliverable events helps to identify current issues and plan corrective actions or even adapt predefined goals. These decisions influence upcoming process activities.

Event-driven: KIPs show an event-driven behavior. Events during process execution can impact activity flows inside a KIP. Events can trigger different directions of process execution. Knowledge workers need to analyze the event outcomes and the relevant situational knowledge to react to their effects.

Constraint- and rule-driven: Businesses are always subjected to many kinds of constraints. Constraints can be fiscal limitations, limited capacities of physical supplies or a restricted budget for a project. These constraints play a significant role for the determination of process steps, process goals and the process execution. Through their decisions, knowledge workers need to meet the requirements of these constraints.

Non-repeatable: Another factor which makes knowledge-intensive processes difficult to manage is that in most cases process plans are not repeatable because process behavior varies across different projects. Although many organizations strive for the implementation of repeatable processes, this is only possible in limited ways. Guidelines, patterns and models can be applied to formal processes or single process fragments with consistent process input. KIPs, however, do not have consistent knowledge as input and hence produce different process outcomes.

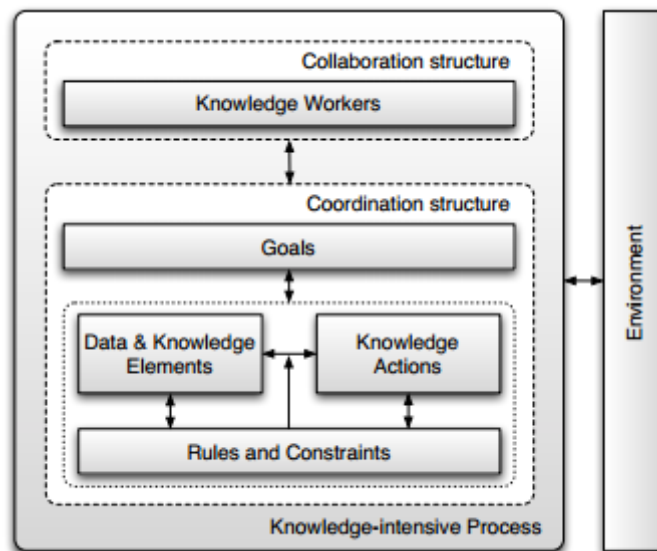


Figure 2.2.: Components of a KIP [12].

These characteristics form the components of a knowledge-intensive process. A KIP consists of a collaboration and a coordination part, which influence each other. The collaboration structure holds the knowledge workers. The coordination structure consists of goals, data and knowledge elements, knowledge actions, rules and constraints. The single components of the coordination structure exhibit relations amongst them. Knowledge actions are carried out based on the available data and knowledge elements. Knowledge actions and knowledge elements are restricted by existing rules and constraints. The three elements data, actions and rules contribute to the achievement of predefined goals. Goals are primarily assessed by knowledge workers of the collaboration structure. Furthermore, KIPs are in permanent interaction with the environment. Figure 2.2 shows an overview of the KIP components and their relations.

2.3. Requirements for Knowledge-Intensive Processes

The different characteristics of KIPs lead to an inevitable analysis on how these processes can be supported. Traditional business process management (BPM) aims to automate pre-specified processes. These are, unlike KIPs, highly structured and repetitive. Performed on an operational level, they rely on the same input and produce a predefined and predictable outcome [51]. All activities, dependencies and data entries are known before runtime. KIPs, on the other hand, are highly dynamic and unpredictable. Hence, BPM approaches are often limited in their support for these new emergent work practices. They are not flexible enough to adapt to changing process actions and contexts.

To better highlight the differences between prespecified and knowledge-intensive processes two real world scenarios are depicted from the health care domain. A prespecified process would be the preoperative procedure. This process is usually initiated with a medical examination including blood tests and ECGs. Then potential risks of the surgery are discussed with the patient. Finally, the patient has to sign a consent form. This process represents a routine procedure for all patients undergoing surgery. An example for a knowledge-intensive medical process is a complex treatment plan for a patient. It varies from patient to patient, is highly dynamic and depends on a number of patient related data and events. This treatment process is also described as a diagnostic-therapeutic cycle as shown in Figure 2.3. Thereby, the physicians represent the knowledge workers who perform knowledge actions, e.g. diagnosing the patient and choosing an appropriate therapy. The patient information represents the knowledge data which influence the knowledge actions of the physicians. The goal of the medical treatment is to cure the patient from his or her disease or distress.

This example of the medical treatment shows that in most cases it is not possible to define a strict set of prespecified activities. Hence, in order to support these kinds of scenarios, it is necessary to determine the requirements to support the components of KIPs presented in the previous section. In the following, these requirements are presented on the basis of the diagnostic-therapeutic cycle.

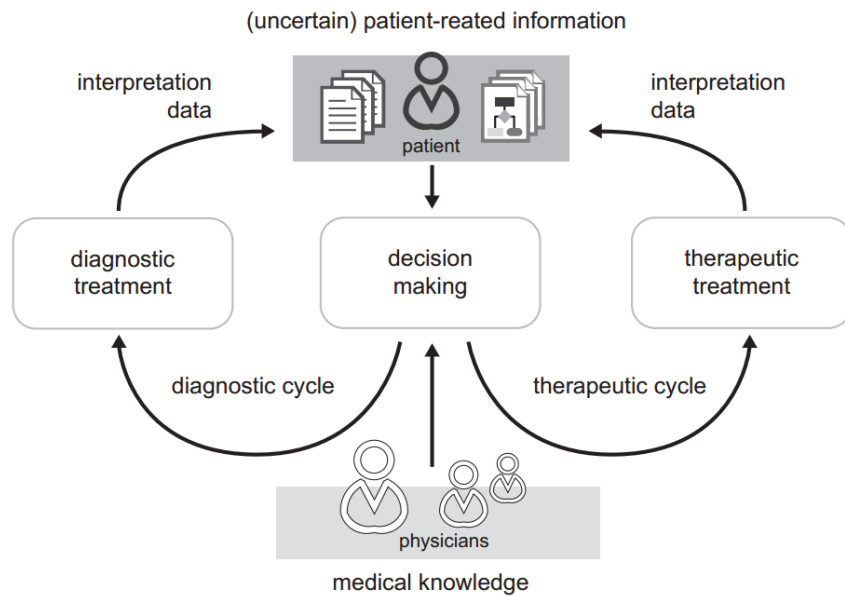


Figure 2.3.: The diagnostic-therapeutic cycle of patient treatment [44]

Requirements towards the knowledge process: *KIPs must be flexible and deal with unanticipated exceptions.* A therapeutic plan is established by a physician based on the patient's health condition. However, the course of therapy can change due to emergent events. For example, if a patient is responding well to a therapy, the physician might alter his therapy plan, e.g. reduce medication. If a treatment does not show any improvements of the patient's health, the physician might even need to change the type of treatment. Hence, it is necessary, that they can adapt or change the methods during the treatment cycle.

Requirements towards knowledge data: *Knowledge data must be accessible and editable at all times. Shared knowledge data needs to be consistent.* The patient information comprises for example demographic data, the current medical condition and the medical history of a patient. This data can alter during the course of disease or therapy and influence the continuous treatment. Therefore, it is necessary to allow access to this information at all times and to collect and update all information. Furthermore, several physicians and medical staff like nurses may have joint access to the patient's data. It is hence necessary, that all shared data between them is consistent.

Requirements towards knowledge actions and goals: *All knowledge actions are knowledge-data-driven. Changed knowledge data must allow the change of actions and goals.* All decisions by physicians for the diagnostic and therapeutic treatment are based on the current patient information. Since the patient's condition can change, physicians need to be able to adapt or change their decision regarding the treatment.

The overall objective of a medical treatment is the relief of distress, the promotion of health

and the cure of a disease. Physicians often set sub-goals, e.g. the first diagnosis or desired outcomes of therapeutic steps. All these goals depend on the patient's health condition. This also implies, that objectives can change during the treatment. For example, in more severe cases goals can be pain relief and promotion of peaceful passing. Hence, physicians need to be able to define, adapt or discard existing goals during the process.

Requirements towards knowledge workers and constraints: *All knowledge workers and knowledge actions must be documented. Knowledge workers must be able to define constraints. Defined constraints must be adhered to.* Physicians must create reports and documents containing medical orders regarding the diagnostic or therapeutic treatments. These documents contain all medical decisions, the medication and conducted medical tests. The documentation is necessary so that the impact of the decisions on the treatment and on the patient's health condition can be determined. Furthermore, it must be documented which physicians and medical staff are responsible for a patient. If several physicians work together on a patient, they need to conduct meetings discussing and sharing the information about the patient and his treatment. A large number of complex laws govern the medical treatment of patients. Decisions and actions of physicians must comply with these predefined constraints. Furthermore, physicians can define constraints themselves, e.g. temporal constraints for therapy duration.

3. Mobile Usability and User Experience

Mobile devices have become ubiquitous tools on both consumer and enterprise level. The time when mobile phones have been mainly used for making phone calls is noticeably passing by. Today, they provide users with a wide range of functionalities replacing everyday gadgets such as clocks, cameras or partly even laptops. They moreover have become a primary link for social networking and searching information on the mobile web. Mobile devices continuously increase in number and variety making research in mobile usability and mobile user experience an important task. Mobile interfaces need to be specially designed in order to achieve satisfying user experience. In this chapter, the foundations of usability and user experience are introduced. First, relevant terms and definitions are presented in section 3.1. Section 3.2 presents general usability attributes. Section 3.3 introduces well-established usability guidelines and norms. Finally, section 3.4 presents principles for mobile user interface design.

3.1. Definitions

Following the massive introduction of digital technology on the market, terms such as ergonomic design and usability are often used when it comes to design decisions. 'User-friendly' and 'excellent user experience' have become popular catch phrases for product marketing. They relate to how well a product is designed for its purpose. Products which provide an engaging experience can have significant advantages in attracting consumers. Thereby, usability and user experience address any kind of device, software or system. Although the terms usability and user experience are often used together, they show subtle differences between them. Experts in this field have frequently depicted that they comprise different scopes of measure, but also share an important relationship. In the following, the definitions of usability and user experience are presented.

3.1.1. Usability

Usability, often referred to as ease of use, is a fundamental concept of user interface design. The term originated in the beginnings of human-computer interaction in the 1980s along with the invention of the graphical user interface (GUI) [37]. Personal computers started to reach a wider audience making technology more pervasive in the workplace and at home. However, interactive computing implied knowledge about the system, making the use of computers a difficult task at that time. Today's understanding of usability comprises a larger set of criteria than it did back then. Several definitions of the term usability can

3. Mobile Usability and User Experience

be found in the literature. Many of these definitions are corresponding with the overall concept, but contain slight distinctions on more specific aspects. They moreover define usability with regard to different views [4]. The definition provided by the ISO 9241-11 standard for Ergonomics and Human Computer Interaction focuses on a user-oriented view, addressing the use of context and usage.

Usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use [34].

Another more detailed definition of usability is provided by the ISO standard 9126 for software engineering:

Usability is the capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions [36].

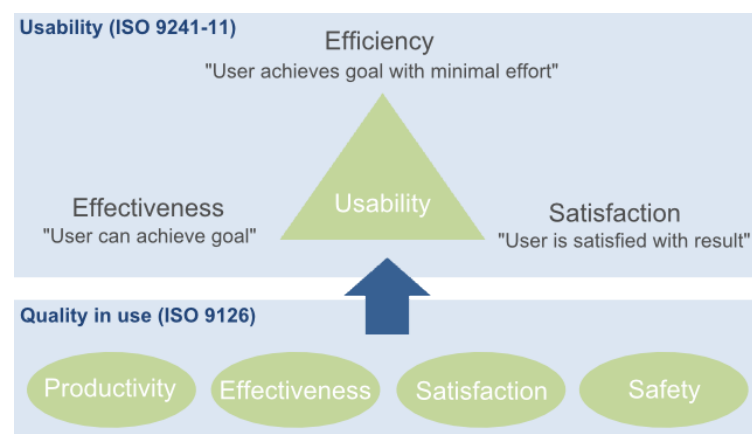


Figure 3.1.: Definition of usability and quality of use.

The definition of usability is complemented with the definition of *quality in use*, stating that users need to achieve specified goals with effectiveness, productivity, safety and satisfaction in specified contexts of use. Both definitions are concerned with human-centered design, also called user-centered design. The user-centered design approach represents an important concept of good usability for all systems. Thereby, user-centered design is "an approach to system design and development that aims to make interactive systems more usable by focusing on the use of the system and applying human factors/ergonomics and usability knowledge and techniques" [13]. The design of a system should emphasize the weaknesses and strengths of the user. The essential goal of user-centered design is to shift the focus from technology to users and user issues and carry out early user evaluations for a successful design [50].

3.1.2. User Experience

User experience design, also UX design, is a fairly new area of discipline. Although it can be linked to usability, it addresses different elements of human-computer interaction. It encompasses all aspects of interaction between a user and the system. While usability addresses the question of how well a user can achieve his goal using the system, user experience also questions how users experience the system while doing so. User experience is directed at the user's perception on an emotional scale. However, there is not one unified and consensual view or definition of user experience. This can be seen as the result of a broad landscape of diverse research and work on this topic. An often used definition found in the literature is delivered by the ISO standard 9241-210. It defines user experience as follows:

User experience comprises a person's perceptions and responses that result from the use and/or anticipated use of a product, system or service.[13].

The ISO approach tries to separate both concepts usability and user experience with their given definitions. The focus of usability lies on the actual situation in which users try to accomplish their goals. User experience also covers effects before and after using a system (see Figure 3.2). Because user experience also takes customer processes that lead and follow the usage of a system into account, it comprises various additional areas such as branding, human factors or even usability.

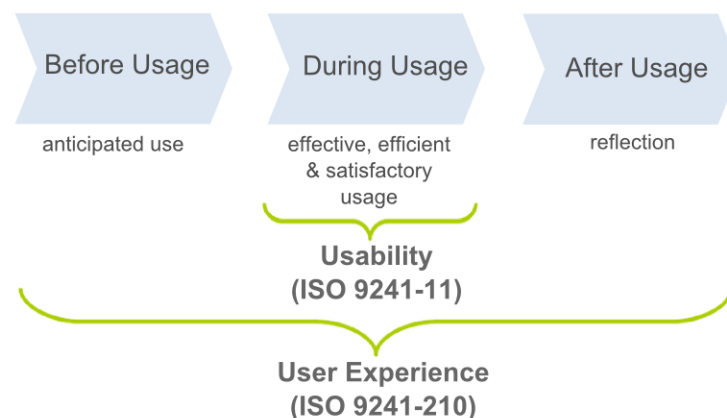


Figure 3.2.: Dimensions of usability and user experience.

3.2. Usability attributes

Usability is often associated with attributes. These attributes represent desirable features of a system which describe how a system should interact with the user. The ISO definition of usability names the three attributes effectiveness, efficiency and satisfaction. Effectiveness looks at the rate of successfully completed tasks. Efficiency addresses the necessary effort and user performance to complete a task and satisfaction comprises the comfort and acceptability of the user[34].

Nielsen identified five key attributes of usability [45]. Thereby, he states that usability is a quality attribute which assesses the experience of a user with the system during interaction. He defines usability in terms of five characteristics:

1. **Learnability:** how fast can a user learn the system during the usage
2. **Efficiency:** how efficiently can the user use the system after he has learned it
3. **Memorability:** how well can an infrequent user return to the system
4. **Errors:** how many errors do users make when they use the system
5. **Satisfaction:** how well do the users perceive the design of the system

During the development process, usability attributes finally become usability requirements. They turn into means of measure which determine how well a system interacts with the user. Hence, they have a direct impact on the outcome of a design and, in the end, the perceived user experience.

3.3. Usability guidelines

Human Computer Interaction experts propose several factors that need to be considered for the design of user interfaces. Hence, many principles, standards and guidelines were formulated over time to promote system usability. They address structural aspects as well as layout and styling. Usability guidelines aim to support designers in their decisions during the design process. Principles are often based on academic research conducted on usability or the experience of usability practitioners. But other areas of research such as communication theory, cognitive psychology and ergonomics did influence many principles as well. Goal of these principles is to unify form and layout with functionality. In this section, sources of guidelines are presented. Well-known design principles and standards are introduced. They represent general guidance for UI design and are applicable to all user interfaces.

3.3.1. Eight Golden Rules by Shneiderman

Shneiderman [58] proposed a collection of principles which he derived heuristically from his research in Human Computer Interaction. The principles are applicable to most interactive systems and are widely acclaimed in the development of websites.

1. **Strive for consistency:** This rule suggests maintaining consistency through the web appearance. This applies to menus, content, commands and prompts. Consistent actions should be called in similar situations. This also applies to layout and style regarding colors and fonts. It lowers the learning curve of users due to the familiarity of a site.
2. **Enable frequent users to use shortcuts.** Shortcuts help to work more efficiently and save time. Especially frequent users rely on keyboard commands since it can speed up the work. Hence, it is recommended to provide abbreviations, shortcut keys, commands and macros.
3. **Offer informative feedback:** Every user action should provide a system feedback. Feedback helps to improve user satisfaction since it can dissolve confusing situations by providing explanations. The type of feedback depends on the performed action. Frequent or minor actions only require modest feedback. For infrequent and major actions the system should provide more informative and extensive feedback.
4. **Design dialog to yield closure:** Dialogs need to provide a clear sequence of actions containing a beginning, middle and ending component. Partitioning dialogs helps to guide the user through the process. After the dialog completion the user needs to be provided with an informative feedback about the status, the outcome and event or progress.
5. **Offer simple error handling.** When users use a website unexpected situations or events can occur which can prevent a normal operation. Typical error sources are data entry forms. Therefore, it is important to design a system where the user cannot make these errors from the beginning. Errors need to be identified and handled by the system.
6. **Permit easy reversal of actions:** As the previous principle points out, users can unintentionally cause errors. The system need to provide a possibility to reverse the actions which led to an error.
7. **Support internal locus of control.** A system needs to give the user the feeling of being in charge. The system needs to respond to the user's action. It should not respond with any unexpected events.
8. **Reduce short-term memory load.** The amount of information a user can save in the short-term memory is limited. Therefore, the amount of information and content on a website needs to be kept in moderation and the layout needs to be simple. The user should learn easily and fast to perform actions.

3.3.2. Nielsen's Usability Heuristics

Nielsen's [45] heuristics are counted among the most applied principles for user interface (UI) design. He proposed overall ten principles for better interaction design. The heuristics represent general best practices rather than specific guidelines. They aim to help usability practitioners and product designers to identify typical trouble spots of UI design and overcome these problems with appropriate methods. In the following, the principles are explained with several examples.

1. **Visibility of system status:** The system informs the user always about the current state through appropriate feedback. Examples are progress bars or feedback messages after user actions.
2. **Match between system and the real world:** This heuristic suggests projecting the mental model of the user about the real world into the virtual environment. The user interface should contain natural language and symbolic which is familiar to the user and appropriate to the type of website.
3. **User control and freedom:** This principle addresses the navigation of a site. Users need to know where they are, how they got to the current page and how they can go back. Nielsen suggests to provide "undo" and "redo" features to reverse user errors without making a long detour.
4. **Consistency and standards:** Similar to Shneiderman's principle, consistency throughout the user interface should be provided. Buttons and items should behave the same way as users may find them on other pages (e.g. a button should be called 'search' instead of 'find this').
5. **Error prevention:** Users should be protected from possible errors due to their actions. Error-prone actions need to be eliminated. This especially addresses forms. They need to contain clear labels and highlight required fields.
6. **Recognition rather than recall:** The system should not burden the user with too much memory load. Information for the use of the system should be easily available to the user. Relevant information should be on the same page. For example, when buying a ticket for a soccer game show the user the different seating areas or categories of the arena in an image.
7. **Flexibility and efficiency of use:** Frequent users should be able to tailor their actions, e.g. by creating macros.
8. **Aesthetic and minimalist design:** Dialogues should only contain relevant information and should be as simple as possible.
9. **Help users recognize, diagnose, and recover from errors:** The system needs to provide error messages which give more insight about the current problem and should offer a solution.

10. **Help and documentation:** If necessary, documentation for the system should be provided as additional source of help. Documentation should be easy to find and concise.

3.3.3. Usability Standards

Standards are developed by the International Organization for Standards (ISO) and the International Electro Technical Commission (IEC). According to the ISO, a standard is "a document that provides requirements, specifications, guidelines or characteristics that can be used consistently to ensure that materials, products, processes and services are fit for their purpose" [33]. The ISO introduced several usability standards which are classified into two different categories: process-oriented standards and product-oriented standards. Figure 3.3 gives an overview of different standards of both categories. In the following, one standard of each category is explained in more detail.

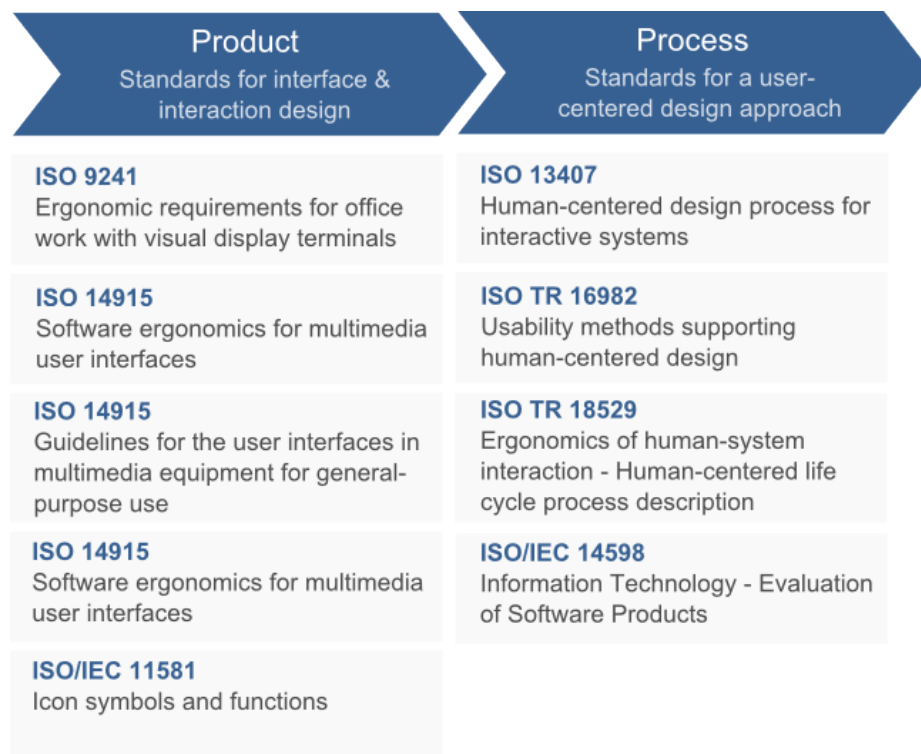


Figure 3.3.: Product- and process-oriented standards.

The ISO 9241 standard is a set of ergonomic principles for office work using a visual display [34]. It thereby focuses on the interaction between the user and the information system. It provides requirements and recommendations towards hardware, software and environment attributes. The most prominent part of this standard is part 10 containing dialog principles. It describes seven dialog principles of ergonomic design applicable to all types of interactive systems. These principles are suitability for the task, self-descriptiveness, controllability, conformity with user expectations, error tolerance, suitability for individu-

alization and suitability for learning.

Another standard which is frequently applied is the ISO 13407, the standard of human-centered design processes for interactive systems [35]. It provides an explanation of activities for user-centered design throughout the product life cycle. According to the ISO, by following this standard an effective, efficient and satisfying system for the user can be developed. It lists several benefits of adopting human-centered design. The user will easier understand and learn the system which reduces costs of training and support. It increases the productivity of users and hence of organizations due to operational efficiency. Finally, the overall product quality and aesthetics can be improved, which leads to competitive advantages over competitors.

3.4. Designing for Mobile Interfaces

For the development and design process of mobile interfaces, it is important to clarify the differences between the desktop and mobile environment. Reaching the usability level of a personal computer on a mobile device is a very difficult task. They are different devices with different purposes. Some of the technical accessibilities of a mobile device can hold back a good user experience. The platform differences of mobile devices and desktop computers make it substantial to create mobile optimized websites. Also user behavior and user expectations differ between the two devices. Mobile users are commonly less forgiving and less patient. They are goal-oriented and expect easy and fast access to a mobile website [47]. Hence, additionally to the previously presented general usability principles, principles for designing usable mobile interfaces are introduced.

3.4.1. Mobile Websites vs. Desktop Websites

A desktop website differs from a mobile website in several aspects. Although both use the same internet sources, the browsing experience differs according to the access device. These differences establish the need for mobile-optimized websites. In the following, the key differences regarding display size, interaction methods and technological capabilities are discussed.

Screen size

The screen size is probably the most visible difference between a mobile device and a native desktop computer. The available screen size decreases with portability. Smaller screens and different screen resolutions can increase cognitive load of the user. A limited space means that less information is visible and less space is available to work with. On desktop computers windows and content can even be displayed side by side. On mobile websites a parallel presentation is almost impossible. Users need to navigate through the website to access different content.

Interaction methods

Another major difference is the way of interaction between user and device. The interaction technique changes from a mouse-driven input to a finger-driven input. The available input determines the available features on a device. For example, the mouse input

provides a hover state and tool tips which cannot be realized on a mobile phone. These elements are less natural for touch input. Furthermore, the real keyboard provided in a desktop environment allows faster typing. On the other hand, on mobile devices, large input over the keyboard can become very cumbersome.

Technological capabilities

As light weight and small devices mobile phones are limited in terms of processing power, memory, browser capabilities, bandwidth and connectivity. All these aspects result in lower performance, higher latency and loading times. While videos, large images and animations can easily be integrated into a desktop site, on mobile websites this can negatively impact page load times. Another factor influencing the loading speed of a website is the cellular network connection. Best case, users connect via a 3G or 4G network. However, the quality of mobile wireless access can deteriorate due to several factors the user has no control of, e.g. geographical reception barriers.

3.4.2. Interface Guidelines for Mobile Devices

To provide the best possible user experience on a mobile platform, Gong and Tarasewich [24] have created several guidelines for mobile interfaces. They adopted four of Shneiderman's rules and extended them with mobile specific principles. Weiss [70] also formulated several principles for mobile interfaces. Both conform in many aspects and complement each other.

The following principles of [24] and [70] are additions to the previously presented usability guidelines of Shneiderman and Nielsen.

- **Design for multiple and dynamic context:** The mobile context is different from the context of native desktop computers. The mobility of handheld devices can lead to changes in the environment, e.g. brightness, locations, noise level. Therefore, different aspects that need to be considered are for example type of input, font size or colors [24].
- **Design for small devices:** As mentioned before, the screen size is a key difference between mobile devices and personal computers. Physical limitations need to be overcome by designing content, buttons and input in appropriate size to guarantee easy operation on small screens [24].
- **Design for limited and split attention:** Mobile users may concentrate on more than one task when using a mobile device. Their main focus may not lay on the use of the mobile device solely. Mobile interfaces need to require as little as possible attention from its users. Visual attention, interaction, input and output need to be designed as simple as possible [24].
- **Design for speed and recovery:** Research has shown that users are less patient when it comes to mobile use [47]. Mobile users are more demanding towards loading times. Hence, time constraints are an important factor for mobile website development [24].

- **Design for top-down interaction:** The top-down approach suggests reducing information overload by providing hierarchy or multi-level mechanisms. This will avoid excessive scrolling and information load [24].
- **Allow for personalization:** Mobile devices are more personal than desktop computers. A mobile device is usually only carried by a single person, while desktop computers can be shared by many users. A user should be able to personalize a mobile application regarding usage patterns, skills and preferences. E.g. users should be able to decide which content to display and which to hide [24].
- **Design for enjoyment:** Joy of use is a rather new quality attribute. Its main concerns are aesthetics and positive user emotions. Especially aesthetics have become an important acceptance factor for users. An appealing design can lead to positive user response [24].
- **Consistency between platforms:** While overall consistency is a main usability attribute, further dimensions of consistency need to be considered for different mobile platforms and devices. E.g. consistencies across mobile browsers need to be ensured [24, 70].
- **Select vs. type:** Input which requires typing can become cumbersome on touch devices. The soft keyboard is less precise and often implies more work and time. Furthermore, the keyboard can hide a large amount of content on the small screen. Therefore, where it is applicable, users should be presented with selection mechanisms, e.g. buttons, checkboxes or dropdowns, instead of keyboard input [70].
- **Clickable graphics should look clickable:** Buttons and clickable icons should always be recognized by the user. This can be achieved through appropriate styling, e.g. high contrast and typical button layout [70].
- **Use icons to clarify concepts:** Icons have become popular design elements for mobile applications. They can enhance the aesthetic appeal and provide additional assistance [70].

4. Related Work

Several researchers have stated the need to support knowledge work of organizations with the use of mobile devices. In this chapter, related work done in the past is presented. In section 4.1 a brief review of published literature regarding mobile knowledge work is presented. In the section 4.2, popular task management applications are introduced, which are currently available on the market.

4.1. Mobile Support for Knowledge-Intensive Processes

Although there has been increased ongoing research on mobile support for business processes, only little research has been done so far which investigates mobile support for knowledge workers. Most of the research conducted in this field focuses on specific elements of knowledge work or knowledge intensive processes. These elements are mainly mobile collaboration, knowledge management and task management.

To determine the requirements towards mobile support systems for knowledge work, several researchers first needed to examine how of knowledge work and the mobile environment intersect. Grimm et al. [27] developed a general concept for the management of mobile knowledge work (see Figure 4.1).

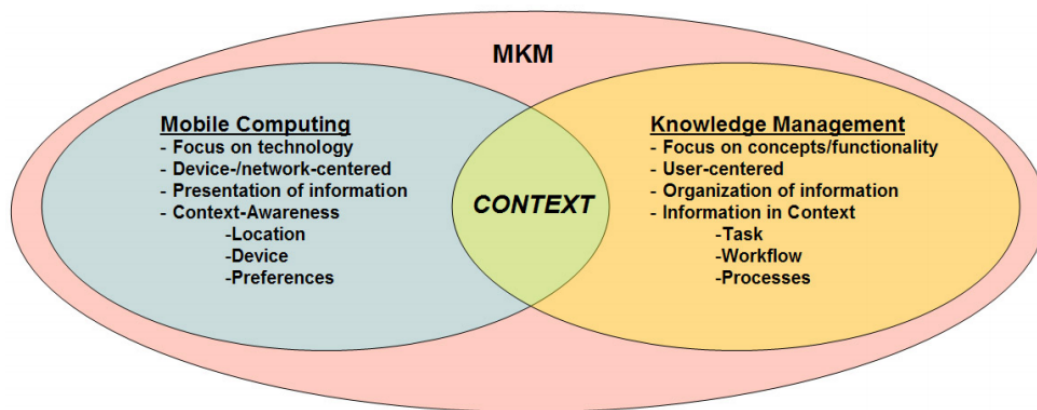


Figure 4.1.: The mobile knowledge management concept by Grimm et al. [27].

Their concept comprises key aspects of the two areas *mobile computing* and *knowledge management*. By analyzing existing systems and user requirements they concluded that context-awareness represents an essential prerequisite for successful mobile knowledge work. Context thereby comprises temporal, personal, organizational and environmental conditions

4. Related Work

of the user. Other researchers have focused on characteristics of mobile knowledge workers [72, 62, 1, 69]. Objectives of mobile knowledge work were presented by Tazari et al. [63]. They state that a mobile support system needs to provide all necessary resources for knowledge creation, distribution and optimized task handling. Furthermore, a mobile system should not distract the user from his actual work by laying too much focus on the technology itself. Finally, they suggest that privacy preferences of users are necessary to define roles and permissions.

A widely found solution for the support of knowledge intensive processes are semantic wikis. Traditional wikis consist of pages which are connected via hyperlinks. The general idea of a semantic wiki is to not simply connect pages via links, but to create relationships between pages through related information and meaning. The data of semantic wikis is stored as formalized information in a database and is readable by machines. This allows users to easily search for related content [56].

Ermilov et al. [17] incorporated the concept of a semantic wiki in their mobile solution. Their presented approach aims to support collaborative knowledge work for expert users. OntoWiki Mobile allows users to collaboratively gather, save and browse data in a mobile application. Data is represented in a hierarchical structure and can be accessed through a tree-like list navigation. It further allows users to define so-called knowledge bases to author and manage information. It was developed using HTML5 together with the jQuery Mobile framework to make it platform independent. Parts of the user interface of OntoWiki can be seen in Figure 4.2.

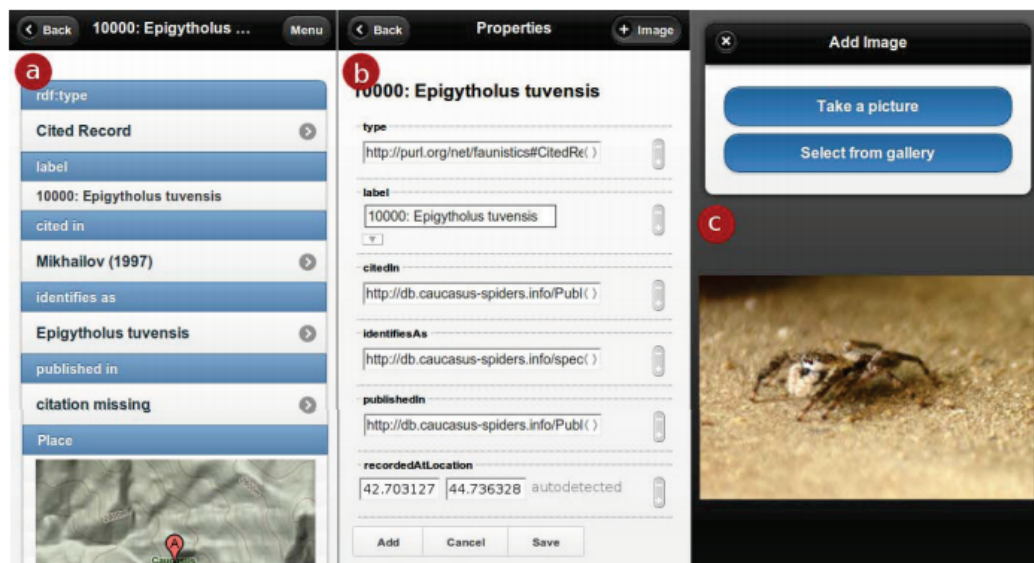
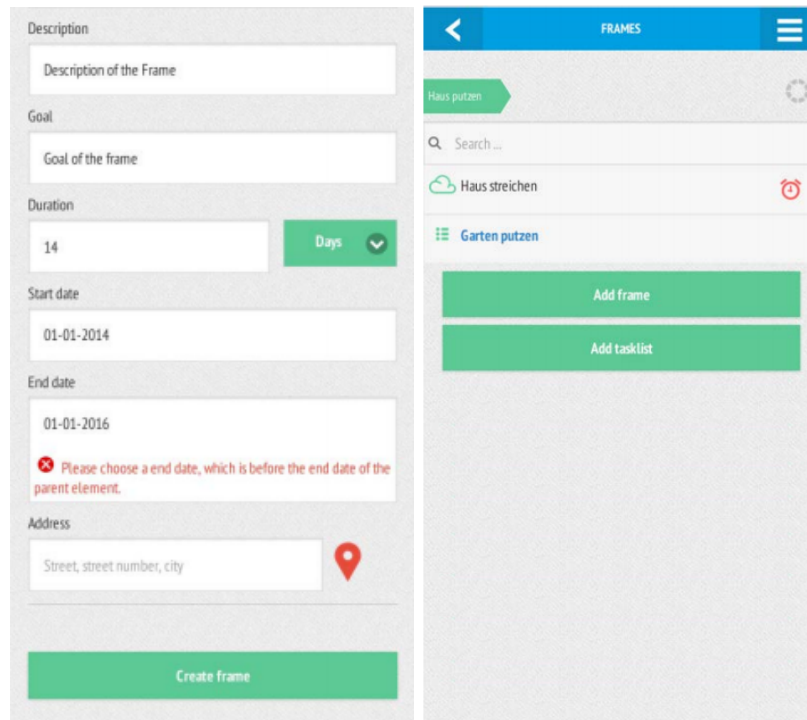


Figure 4.2.: OntoWiki Mobile. a) Overview of meta data. b) Editing of meta data. c) Adding images [17].

The proCollab project of the University of Ulm aims to develop a solution to support collaborative knowledge work based on the KIP life cycle presented in chapter 2. Geiger [22] and Gerber [23] developed task-centered mobile solutions for the proCollab concept. Their

concepts relies on the creation of checklists as reusable task templates. Knowledge is structured in organizational frames. An organizational frame represents for example a project and contains information about responsibilities and work allocations. Figure 4.3(a) shows the interface to define such a frame. The user can define frame goal, duration and set start and end date. Inside such an organizational frame users can define tasks as instances of a checklist. Figure 4.3(b) shows an overview with an organizational frame and a checklist instance.



(a) Defining a new frame.

(b) Overview of a frame and a task.

Figure 4.3.: proCollab.

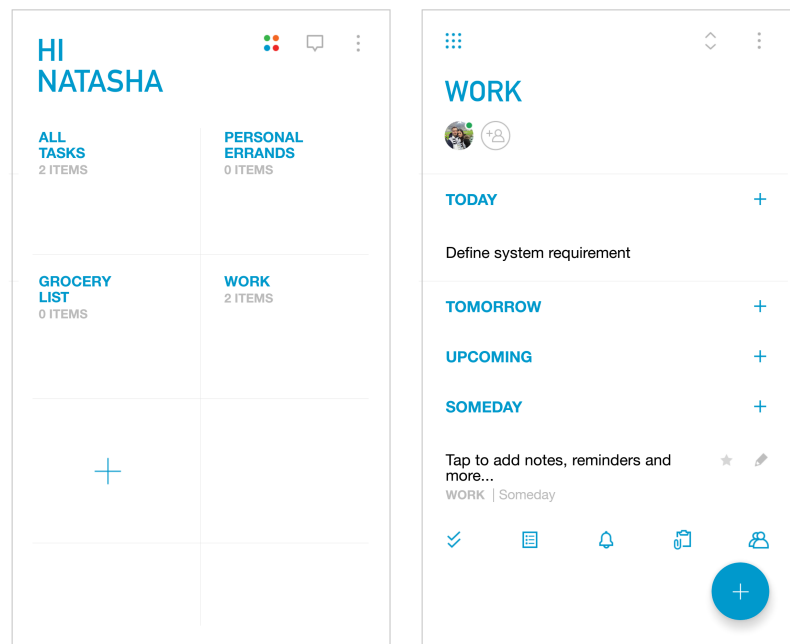
Several other publications address different aspects of mobile knowledge work. Some of them introduce semantic systems for collaborative authoring in a mobile environment [71, 60, 3]. Others deliver solutions to support the collaborative aspect of mobile knowledge work [54]. However, the reviewed solutions differ in several ways from the approach in this thesis. They address mostly specific target groups or expert users. Furthermore, these semantic wikis do not support the creation of tasks which is one of the most relevant use cases of the web application developed for this work. Other missing components are user roles, attributes of tasks and user profiles. Finally, none of the reviewed mobile applications focuses on the usability aspect of their proposed solution. This, however, plays a significant role for user acceptance, satisfaction and most importantly the successful achievement of user goals.

4.2. Task-Centered Mobile Applications

The main use case of the developed mobile website is the management of tasks. Therefore, after reviewing existing literature on mobile support of knowledge work, an overview of mobile applications for task management is presented. These productivity applications are currently in high demand in app stores. They aim to help smartphone users to organize, prioritize and categorize their tasks. Overall five applications are presented, which are currently available as free and premium versions in app stores.

Any.Do

Any.Do is one of the most downloaded to-do-list apps on Google Play and the iTunes Store. The application lets users create categories to organize tasks. Each category gives an overview of upcoming tasks. Users can define notes, add files to task and set location-based reminders. Furthermore, tasks can be split into subtasks and assigned to other users. Any.Do provides three different task overviews: a date view, a list view and a priority view. Furthermore, it offers a daily planner, called "Any.do Moment" which gives users a task overview of the current day. Finally, Any.Do has a browser extension for Chrome and a free web version, which allows users to sync lists between devices and platforms. Figure 4.4 shows the category and task overview of the Any.Do application.



(a) Overview of categories.

(b) Overview of tasks.

Figure 4.4.: Any.Do.

Wunderlist

Wunderlist is another to-do-list app which is available for iOS, Android, Windows Phone and as a web application. The user can create lists which are similar to Any.Do's categories. Inside those lists the user can define new tasks. For each task, the user can set a due date and a reminder. Subtasks, notes, files and comments can be added to a task as well. Unlike in Any.Do, Wunderlist lets users assign a complete list of tasks to other users. Another feature is the use of hashtags. By using hashtags users can assign properties to tasks and manage contexts. Users can then search for hashtags to find related tasks with the same hashtags.

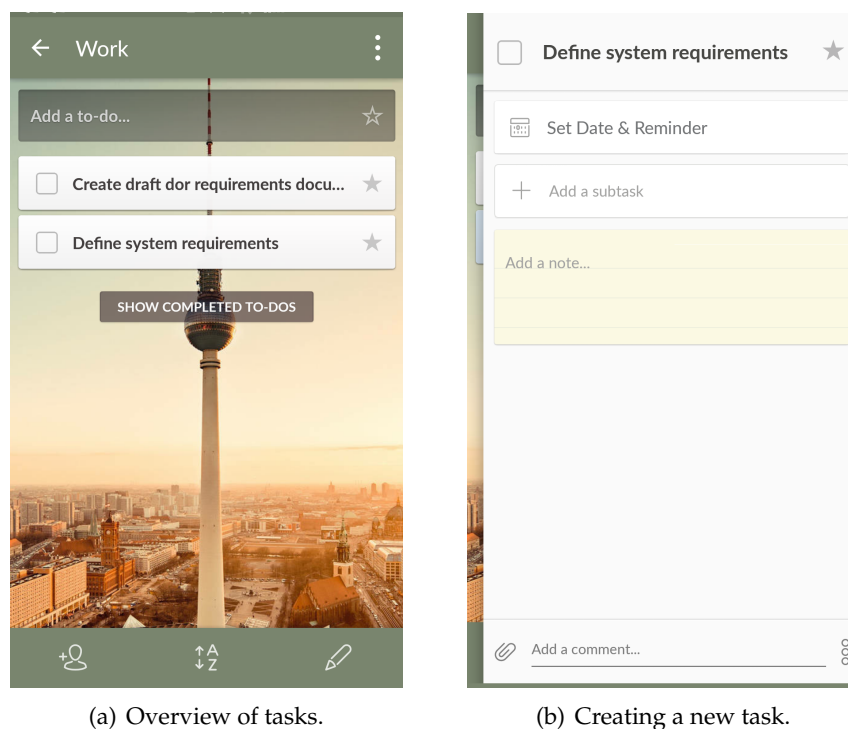


Figure 4.5.: Wunderlist.

Trello

Trello is a collaborative project management tool with a wide set of features. It is based on the concept of cards. Like in Any.Do or Wunderlist the user has different categories called boards. Each board comes with the three initial lists "to do", "doing" and "done". The user can redefine those lists or simply add new ones. In every list, the user can create cards. Cards are similar to tasks. They have a name and a description. The user can also add comments to a card. Trello lets users set a wide range of meta data for a card. They can for example define labels, set a due date, attach files, add checklist items or assign a

cards to other users. Additionally, in Trello the user can see the complete activity history of a card. Cards can be dragged around different lists. The three initial lists for example represent the different development steps of a card.

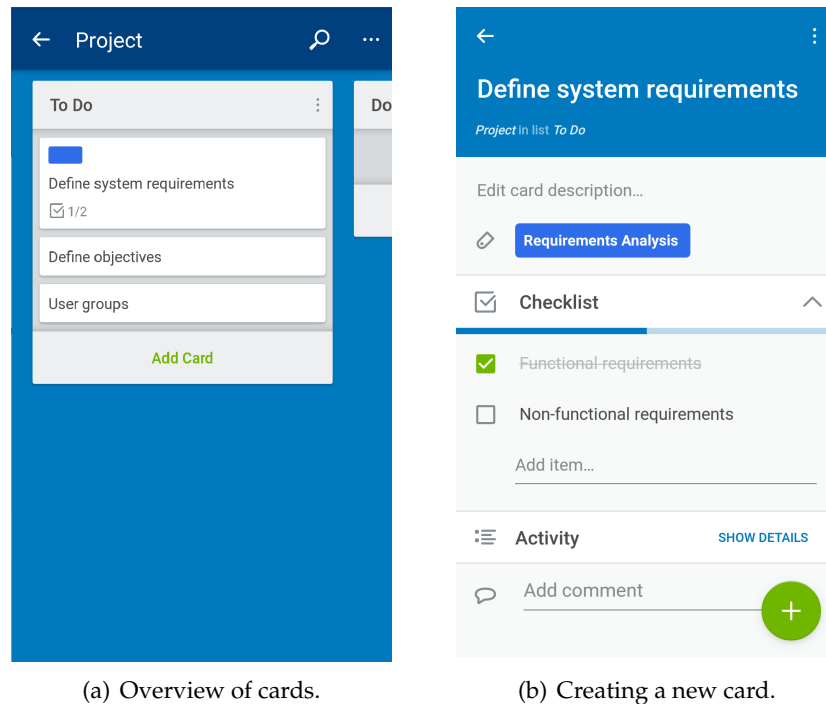


Figure 4.6.: Trello.

Todoist

Todoist is a feature-rich application to manage tasks. It provides an application for mobile and desktop use as well as browser and email plugins. The free version of Todoist comes with a limited set of features. Most of the essential features are only available with the premium version. On the free version users can create projects and add tasks to them. Tasks can be divided into subtasks and defined with due dates and flags which describe the urgency of a task through different priority levels. To collaboratively work on tasks the user can share projects with other Todoist users. Projects can be color coded to differentiate between them. The user can switch between different task overviews, e.g. the current day's or week's tasks. With a filter users can search for specific types of tasks, for example tasks assigned to them or tasks of a specific priority level. A user profile summarizes the number of completed tasks over different time spans. On the premium version users can additionally add reminders to tasks, write comments and add labels. Todoist is a cloud-based applications which allows to synchronize tasks through different devices.

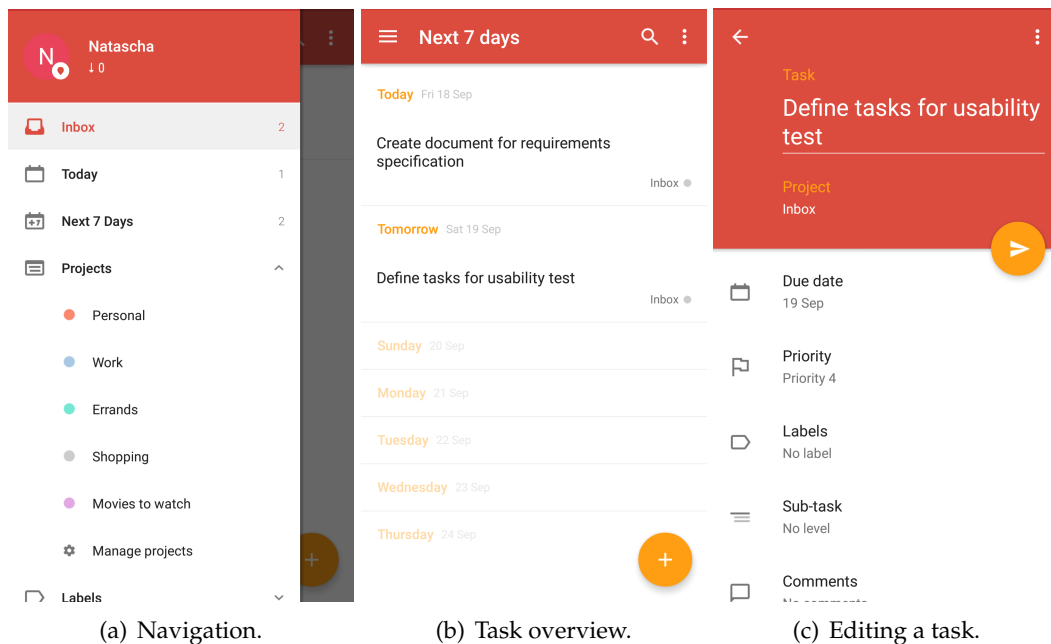


Figure 4.7.: Todoist.

Although the presented mobile applications are well suited for the management of simple tasks, KIPs do exceed their capabilities. The applications show drawbacks in several aspects. The most significant disadvantage is that all applications do not provide extensive knowledge repositories as seen in Darwin. Tasks or projects can only be annotated with short descriptions or comments. They, however, do not provide the necessary means to manage large amount of information or knowledge. The second main disadvantage is that these applications do not provide the possibility to create and maintain reusable work templates. For every project tasks have to be created from scratch. Although KIPs are diverse, reusable and adaptable work templates can help to decrease the administrative and management efforts. Other drawbacks which are aimed to overcome with the presented solution in this work are limited customization and organization options, limitations of cross-platform availability and easy tracking of the project progress.

Part II.

Conceptual Design

5. A User-Centered Design Approach for Mobile Websites

In this chapter, the user-centered design approach for the development of Darwin Mobile is presented. Section 5.1 describes the process which was applied for the development of the system. The analysis of users and the context of use is presented in section 5.2. A requirements analysis comprising functional and design requirements is discussed in section 5.3. Section 5.4 describes the design process of creating the initial design solutions for the user interface. Finally, in section 5.5 a brief overview of the assessment of mobile technologies is presented.

5.1. User-Centered Design

The main methodology to develop the mobile website follows a user-centered design (UCD) approach. Therefore, in this section the fundamental principles of user-centered design are briefly introduced before presenting the development approach for Darwin Mobile.

5.1.1. Key Principles

User-centered design originated in the 1980s with the goal to achieve maximum user satisfaction by involving users in the design process [28]. It comprises collaborative methods to obtain ongoing feedback of users in order to develop systems which meet the users' needs. Therefore, a thorough understanding of users, their environment and their goals is necessary. A broad spectrum of concepts is introduced in the literature on how to apply a user-centered design approach. Three general principles of UCD are (1) the early focus on users and tasks, (2) empirical evaluations and (3) an iterative design process [50]. Integrating users early in the development process helps to understand real-world interaction issues and refine the system design before committing to a potentially insufficient system.

Figure 5.1 depicts the user-centered design process defined in the ISO 13407 standard. It provides guidance throughout the development life cycle. It consists of four main phases which encompass different key activities. Each phase strives for user involvement with appropriate empirical methods. The first phase aims to identify the people who will use the product, their context of use and the goals they want to achieve by using it. In the second phase, the user and system requirements are defined based on the characteristics of the identified target group. In the third phase, design solutions of the system are developed. The last phase comprises the evaluation of the design solutions. The UCD process will be repeated until the system satisfies all user requirements. A variety of methods to collect information about the users and their domain are available. They highly depend on the

type of information to be gathered. Exemplary user data gathering techniques are interviews, focus groups, observations and usability evaluations. A successfully implemented user-centered design can enhance the overall usability of a system and lead to positive user perception.

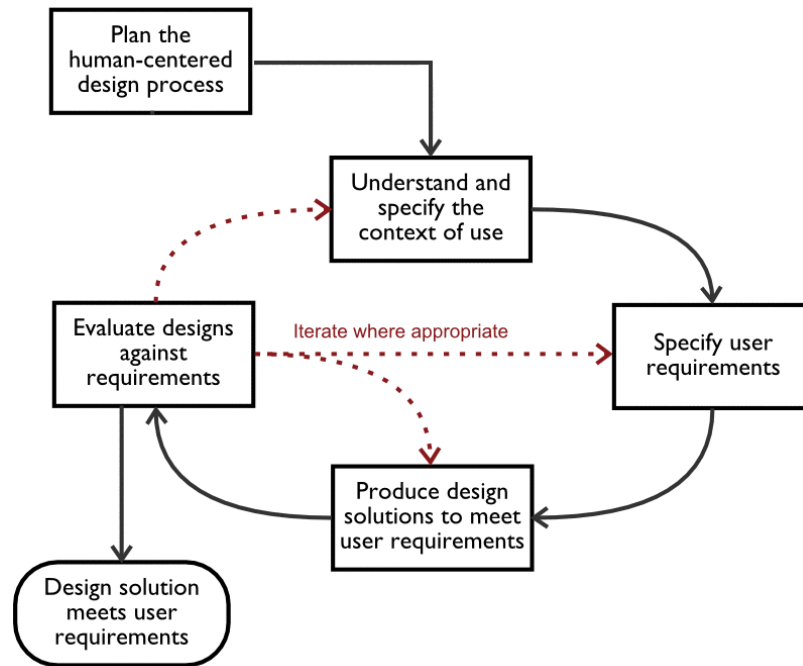


Figure 5.1.: The user-centered design process according to ISO 13407 [35].

5.1.2. Design Process for Darwin

The user-centered design process for Darwin Mobile is based on the UCD process of the ISO 13407 standard. The design process applied for the scope of this work comprises seven steps and is illustrated in Figure 5.2. Each step represents a chapter of this thesis. The first step is the analysis of existing research and tools for the mobile support of knowledge-intensive processes. The results are found in chapter 4. The next four steps are presented in the following sections and comprise the conceptual design of the system. Step 6 represents the implementation which can be found in chapter 6 and 7. Finally, a user evaluation was conducted (see chapter 8) to test the developed system against the specified requirements. Interviews were conducted after creating design solutions to further refine user requirements and improve the design solutions.

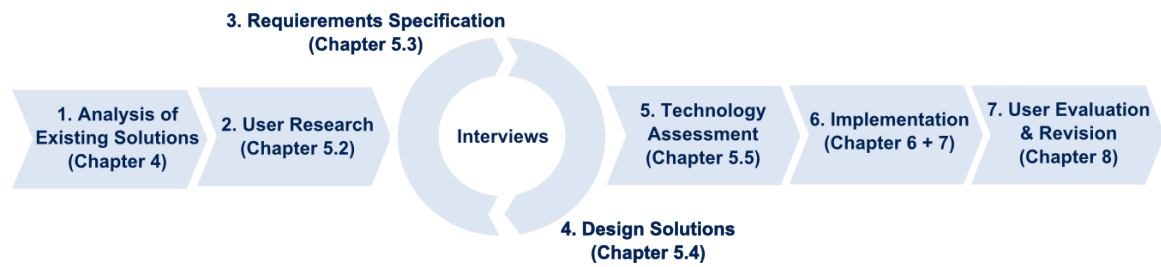


Figure 5.2.: User-centered design process for Darwin Mobile.

5.2. User Analysis

Identifying the target user groups and understanding their requirements towards the system is a substantial task of the design process and crucial to system's quality and success. It helps to determine which functionalities are most beneficial to the end-user as they relate to the requirements the system is designed upon. Therefore, the first step is to understand as much as possible about the users, their work and the context of their work activities so that the system can be built to meet their goals. The user analysis gathers user characteristics and user tasks based on the different user groups. Furthermore, the mobile environment in which the users will perform their tasks is analyzed. The results of the user analysis establish the basis for the requirements specification in the next section.

5.2.1. User Groups

The structuring of knowledge-intensive tasks, e.g. with information system tools, can be performed by two different groups of knowledge workers: modeling experts and non-expert users. Modeling experts have extensive knowledge about modeling languages and methods. They create predefined work and process templates which are later applied during the process execution. Non-expert users are users which take part in the process execution but have limited scientific and modeling background. They also face the task of structuring KIPs, especially during the process execution. Due to the unpredictable character of KIPs, predefined processes of modeling experts often need to be restructured and adapted during runtime. However, this represents a difficult task for non-experts because modeling languages are often too complex and abstract for them [30, 44]. Therefore, special methods are required to support them in this task. Figure 5.3 provides an overview of the two user groups and their characteristics.

Therefore, the main target group of Darwin Mobile are non-experts. Goal is to support them in structuring KIPs without the required knowledge of process modeling and process notation. The mobile application is complemented by the desktop version which provides additional functionality to support modeling experts. They can specify predefined work templates and improve work templates created by non-experts. Darwin offers a Case Management Model and Notation (CMMN) editor to define dependencies. For the mobile application the experts will be disregarded since it can be assumed that a mobile device does not provide the appropriate environment to execute complex modeling tasks.

Hence, the following information about the users of the system can be summarized: end-users are mainly non-experts with limited modeling capabilities. They need to be able to structure processes with simple elements such as tasks. Structuring processes often takes place in a collaborative manner. Therefore, appropriate methods need to be provided to support collaborative work.

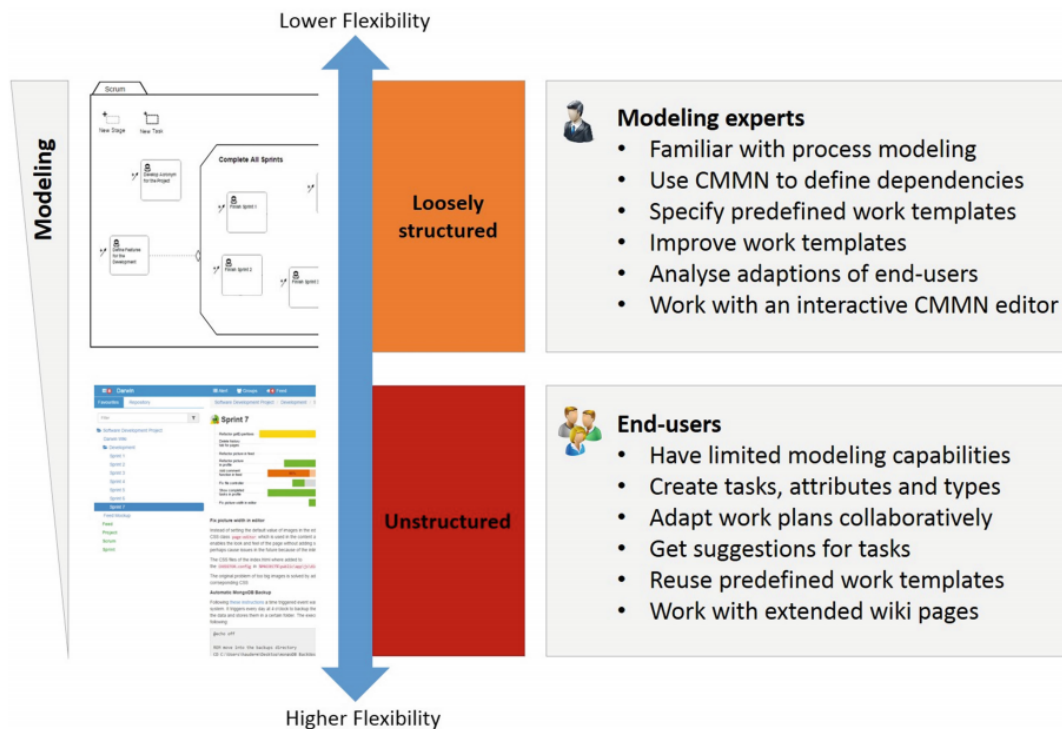


Figure 5.3.: User groups of Darwin and their characteristics [30].

5.2.2. User Environment

Understanding the context in which user activities are performed helps to identify possible technical and environmental constraints. The mobile environment bears several different characteristics compared to a desktop environment. The shift to mobility and the resulting changes of work constraints require a deep understanding of the nature of mobile workspaces. Using a mobile device restricts workers from access to their usual technological and informational resources, e.g. documents or computers. Co-workers also represent informational resources which user are deprived of due to the absence from the office. Therefore, collaboration is a vital feature of efficient mobile knowledge work. Mobile users, moreover, need to be able to track activities of other co-workers to coordinate tasks and prevent information deficit.

Mobile workers themselves can be categorized into two different groups: mobile knowledge workers and mobile field workers. Several differences between them can be found in the literature [72, 62, 1, 69]. Mobile knowledge workers need to make high-level decisions and execute complex, unstructured and unpredictable tasks. This different work environment requires the availability of in-depth information at all times. Further differences of mobile knowledge workers and mobile field workers are listed in Table 5.1.

	Mobile Knowledge Worker	Mobile Field Worker
Composition	Unstructured	Structured
Occurrence	Irregular	Reoccurring
Complexity	High complexity	Less complexity
Urgency	Less urgency	High urgency
Place of work	On the road	On-site
Impact of decision	High impact	Low impact
Information	Deep information (knowledge for decision support)	Location based information (ad hoc information)

Table 5.1.: Differences between mobile knowledge workers and mobile field workers [62].

5.3. Requirements Specification

Based on the results of the user research, a requirements analysis is conducted to identify the activities users will perform on a mobile device to structure KIPs. It will be determined which features are suitable for the mobile use case and which information users will need to access in order to execute their tasks successfully. Use case modeling is applied to describe the functional requirements of the system. It is an effective method to define the interaction between user and system and helps to determine the final feature set. Additionally, scenarios are used to visualize usage examples. This requirements specification describes the final functional and design requirements retrieved in an iterative process together with the design solutions.

5.3.1. Organizing Content

The management of knowledge as content is an important factor since it defines the basis for knowledge generation and transfer. Organizing knowledge and information with the right methods helps knowledge workers to retrieve information easily and apply it efficiently where needed. Wikis have become promising tools to structure and manage

knowledge in organizations [68, 38, 67]. They serve as knowledge repositories with a set of linked pages which are created and maintained collaboratively by users. For the mobile website the concept of a wiki is applied to help users structure their information and knowledge data. Every wiki page consists of unstructured informational text. Users can create and edit informational text on the wiki pages. The use cases for organizing content in wikis is illustrated in the use case diagram in Figure 5.4.

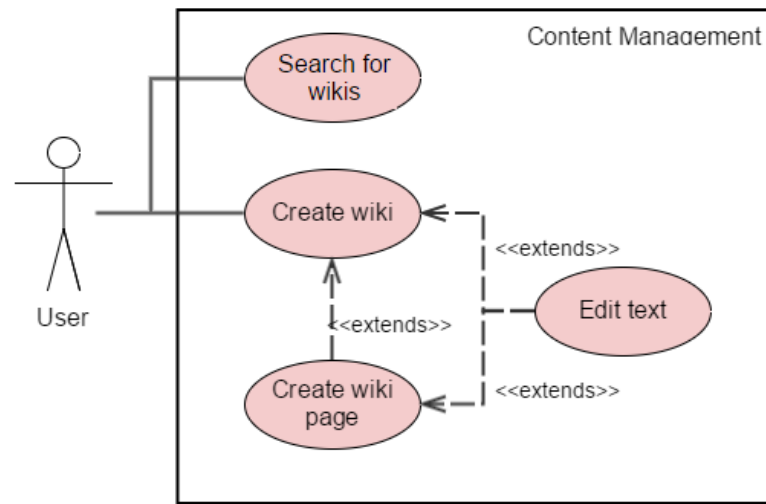


Figure 5.4.: Use cases for organizing content with wiki pages.

A hierarchical structure for the wiki is provided to organize content with superior and subordinate pages. A user can create a wiki with multiple wiki pages in a tree-like hierarchy. Instead of using wikis just as information repositories, a hierarchical structure can also be used to realize a sequential execution of processes. Hierarchical wikis can become very extensive and difficult to query through. A site search is therefore necessary to help users easily find the pages they are looking for. The following scenario depicts how users can create wikis for their processes.

Scenario 1: Project planning

A team of developers kicks off a new web application project. For the project management the scrum methodology is applied. The team creates a new wiki called *Web Application Project*. It contains a small description of the project. Inside this wiki, several wiki pages are created each representing a single sprint. For each sprint the team documents results and work done in form of text on the corresponding wiki page.

Since KIPs are highly data-driven, different types of data can emerge during the process execution in addition to wiki text. While wiki text represents unstructured data, structured data can be found in form of documents, process variables, process results, constraints or any other type of artifact. To represent this kind of structured data *attributes* are introduced. Users can complement wiki pages with data in forms of attributes. Attributes would be for example uploaded files, dates or even other users. The use cases for managing attributes are shown in Figure 5.5.

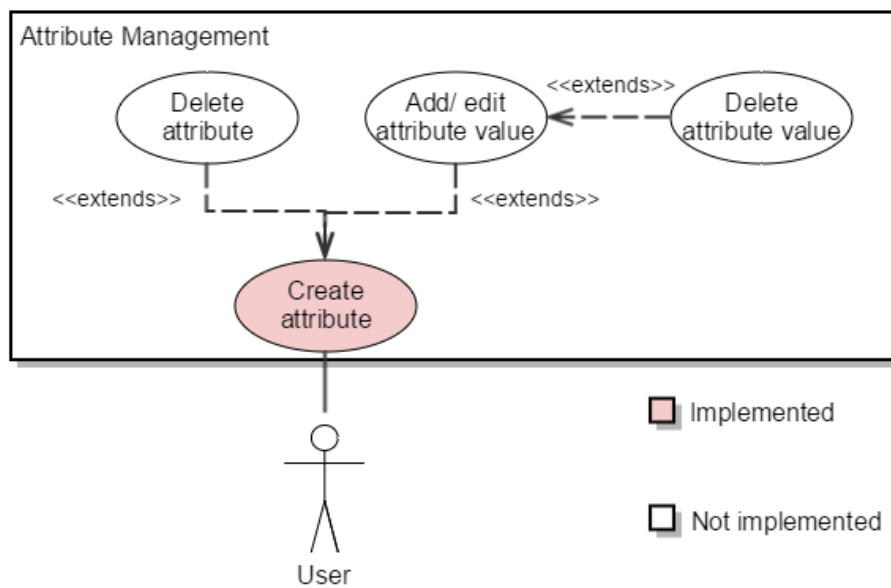


Figure 5.5.: Use cases for managing attributes.

Colored use cases were integrated into the feature set of the develop website, while non-colored use cases were discarded for the scope of this work. Together with an expert user of the Darwin application the decision about which features to drop for the mobile solution was made. The not implemented features can still be used on the desktop version of Darwin. For now, users can retrieve an overview of all attributes of a wiki page and create new attributes. They can define an attribute name and an attribute type. An attribute can be of one of the following eight types: String, Integer, Boolean, Date, Enum, File, Page or User. The previous scenario is continued to show how attributes of different types can be used.

Scenario 2: Sprint planning

The web application team starts planning their first sprint. Therefore, they define backlog items. Backlog items represent the work that needs to be done within a sprint. The goal of this sprint is the delivery of a requirements specification document on a specific date. An attribute named *Due Date* is created. It is assigned the type *Date* and the attribute value is the delivery date of the requirements document. The attribute *Scrum Master* is of type *User*. It contains a link to the profile page of the scrum master of the team. To upload the requirements document an attribute of type *file* is created called *Final Document*. The uploaded file represents the attribute value.

5.3.2. Structuring KIPs with Tasks

Currently, a well-established solution for businesses to structure their processes is the use of a process management system (PMS). These systems define models which describe the process execution from a control-flow perspective, meaning that a process is structured in terms of tasks [40]. The tasks describe the activities which will be performed sequentially or simultaneously. Tasks can be used to coordinate work and create an organizational order of execution. A PMS generally defines models with the assumption of recurring tasks. However, KIPs are rarely or only partly repeatable and the use of a PMS is not sufficient enough. Also tasks often change during the course of process execution. Therefore, knowledge workers need to be able to define tasks and adapt them if necessary during runtime.

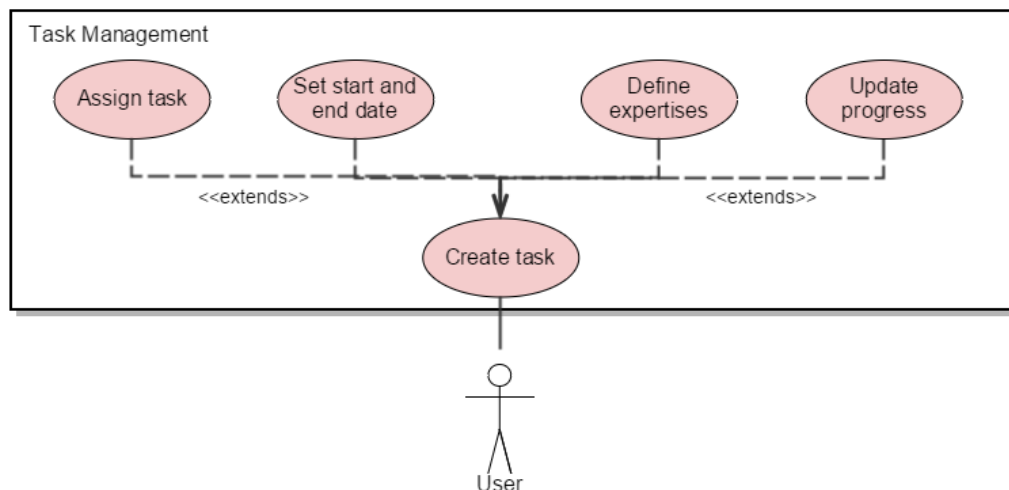


Figure 5.6.: Use cases for managing tasks.

The presented solution allows users to create tasks mapped to wiki pages and alter them at any time. They can define additional task properties such as start date and end date to

determine the order of execution. To keep track of the current task status users can adjust the task progress. Knowledge is often diverse in organizations and not all workers are equally qualified to complete tasks. Therefore, to efficiently utilize different capabilities of workers users can assign tasks to other expert users. Furthermore, they can tag tasks with multiple custom expertises to indicate which skills are necessary for the completion of the task. Figure 5.6 gives an overview of all use cases related to the task management.

Related with the completion of tasks, workers often create output in form of artifacts. The association of tasks and page attributes represents a possible solution to directly map artifact delivery to task completion. However, since delivering files, documents or similar artifacts over a mobile device represents a rather infrequent use case, the mobile solution will for now not provide this feature. Users will be able to assign attributes to tasks on the desktop application of Darwin. Setting the value of an assigned attribute automatically updates the task progress.

5.3.3. Enhancing Visibility of Work Activities

Knowledge work highly relies on collaboration and coordination. Therefore, it is necessary to provide a workspace where workers can easily share their thoughts and impressions with other workers. They also need to be aware of the activities of their co-workers to improve coordination of collaborative work. A popular approach is the integration of social media aspects. Social networks are embraced more than ever in businesses. They are used as information platforms and help to engage communication with co-workers through conversations. Another positive impact of social media is the improvement of the visibility of work activities. KIPs have an intangible character making it difficult to track the actions of knowledge workers. Social collaboration is integrated into the mobile solution in form of a feed. The feed gives an overview of all user activities. User can create new discussion posts and comment or like feed entries. Additionally, a filter will be provided to reduce large amount of data and help users to easily find entries of a specific type. The use cases for the feed are illustrated in Figure 5.7.

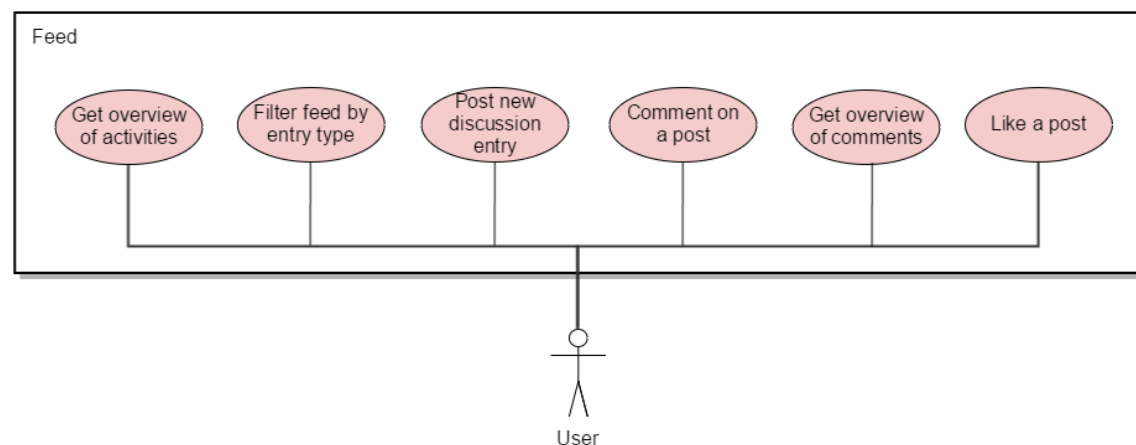


Figure 5.7.: Use cases for the feed.

To further enhance visibility and highlight active tasks, task overviews are introduced. They represent lists of tasks grouped by different periods of time, e.g. tasks for the current day or upcoming week. The user will be notified of overdue tasks with an alert and can see all due tasks in one overview. Figure 5.8. shows additional use cases for better visibility.

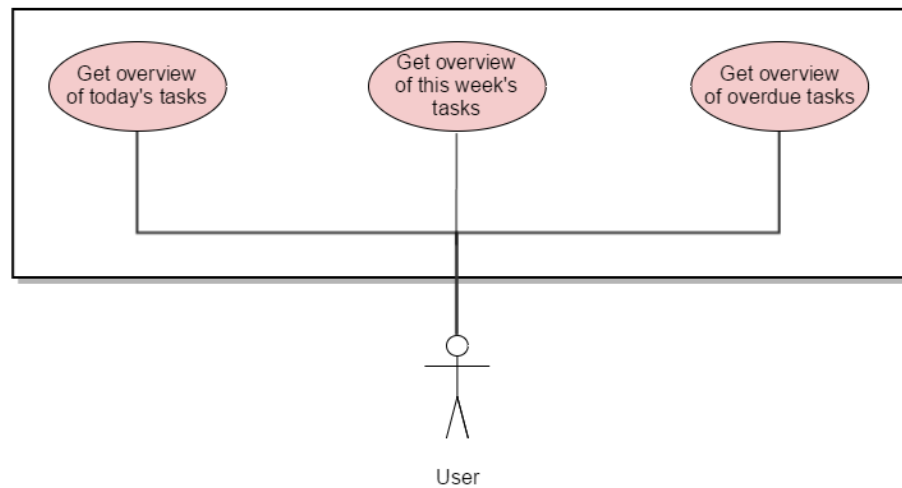


Figure 5.8.: Use cases for enhanced visibility.

5.3.4. Managing Users

Besides the collaborative aspect of knowledge work, knowledge workers also need to be provided with some kind of personal workspace. While the feed aims to provide social context information, a personal profile page aims to gather user related information. The profile page gives an overview of the user's personal information and his task activities.

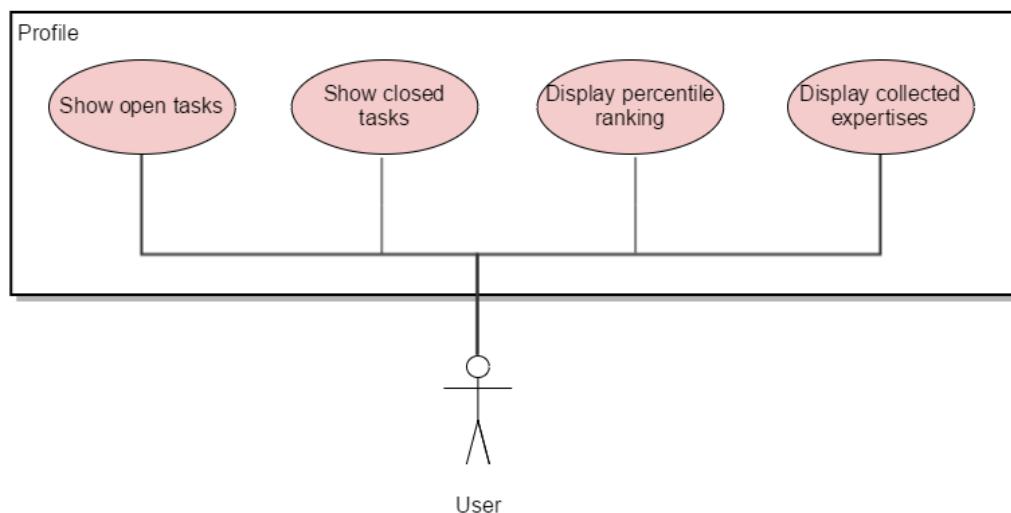


Figure 5.9.: Use cases for the profile page.

The user can track all his open tasks at one spot. Furthermore, he can see all the tasks completed by him. The profile comprises besides the task activities information about the user's skills and capabilities in form of expertises. An overview of expertises documents their skill base. When users complete a task which has expertises assigned to it, the user will collect these expertises in his profile. This creates further incentives for users to increase their work contributions. Another encouragement represents the percentile ranking. It compares the user with other users in regard to completed tasks.

The profile page represents also a way to get a better insight about other users. If a task for example requires a specific set of qualities, the appropriate user for a task can be determined by his expertises. Or if a user has trouble with a task he can reach up to a person who already completed a similar task.

5.3.5. Design Requirements

Since the website is created for a mobile device it is important to define requirements towards the design. A set of requirements was identified which is based on the usability attributes, the differences between mobile devices and desktop computers and the design guidelines and principles presented in chapter 3 of this thesis. They define how the system needs to be designed so it is usable and useful for the target customer. They are categorized in requirements for the design for small screens, the information structure and the feature set.

1. Designing for the small screen

Requirement R01: *Scale down the website to fit the screen.* The mobile website shall automatically be adjusted to the screen size of the device. Scaling down websites to fit the screen prevents horizontal scrolling. This can be achieved for example by setting the viewport through the HTML meta tag. It will create a native app like experience. Another possible solution are media types. Media types render different stylesheets according to the device's screen size [26, 9].

Requirement R02: *Keep content short and simple.* The website shall display information in a short and simple format. The small screen size limits the amount of information visible for the user at first glance. This will prevent increased scrolling and provide a certain reading comfort for the user. This applies to all kinds of content including texts, dialogs and forms. User will not be willing to browse extensively for the content or fill out time consuming forms [9, 47, 19].

Requirement R03: *Prevent the need to zoom.* The system shall not force the users to zoom in and zoom out content. Although more information can be fitted into the display if kept small, this would require the user to zoom in and out of the interface. However, this would disrupt the flow of text and hence the reading comfort of the user. Therefore, it is necessary to display information large enough and provide high contrast [20].

Requirement R04: *Appropriate use of back-buttons.* The website shall fulfill back-button expectations of the user with appropriate response. Back-buttons can represent a source

for errors, e.g. broken links or a redirection the user was not expecting. Users use back-buttons differently. Some use it to go back in their browsing history. At other times, they use it to revert their actions. Therefore, to prevent errors, it is important to consider for every dialog the probable intention of the user when clicking on the back button [47, 9].

2. Mobile information architecture

Requirement R05: *Prioritize your content.* The website shall foremost display relevant information. The most relevant content needs to be immediately accessible to users. Complete content of the desktop website should not simply be minimized for a smaller device. In contrast, content for the mobile website should be selected by having the context of use and user needs in mind [47, 19].

Requirement R06: *Move secondary content to secondary pages.* Since primary content should be available at first hand for users, secondary content should be moved to secondary pages [47].

Requirement R07: *Cut unnecessary media.* Leave unnecessary images, animations and videos to improve performance and page loading times [9, 20, 42].

Requirement R08: *Optimize images.* The system shall provide mobile-friendly images. Images should be optimized for shorter loading times through resizing, compressing or choosing an appropriate image format [9].

Requirement R09: *Give feedback about the loading state.* The system shall provide feedback for the users regarding the loading state of the website. This can be achieved by including progress bars. This will lead to the illusion of shorter waiting times for the user [47].

3. Mobile feature set

Requirement R10: *Provide progressive disclosure.* The system shall present the user with the most relevant functionality first. Users should be presented with the basic features and options. Only offer a larger set of options or features if explicitly requested by the user [47, 8].

Requirement R11: *Reduce the feature set.* The system shall provide the features which are relevant for the mobile use case. This helps to decrease the complexity of the website. Although a wide feature set is often associated with more capability, sticking with only essential features lets users focus better on important tasks and hence contribute to usability. Rust et al. [53] discovered in their research that users prefer usability over a large set of features. To determine the relevant features for the mobile context, user needs must be determined and prioritized.

5.4. Design Solutions

This section briefly describes the design process for the mobile website. The goal was to transfer the functional and design requirements specified in the previous section to a possible design solution. The design process thereby was executed in several iterations. Mockups were created to determine information structure, content and functionality of the mobile solution. Also the appearance of the mobile website and visual details were incorporated into the mockups. To continuously improve the design solution the mockups have been evaluated in interview sessions with experts and users. Several sessions were conducted incorporating the feedback and adapting the feature set. The design of the final implementation relies on the mockups and is presented in chapter 7.

5.5. Assessment of Mobile Technologies

The last conceptual phase of the applied user-centered design process is the assessment of mobile technologies. Before starting with the implementation of the design solutions, a basic understanding of existing mobile technologies is necessary. In this section, the procedure for the assessment of different mobile strategies and frameworks is presented.

5.5.1. Mobile Website vs. Native Application

When developing a system for a mobile device the first question that arises is which mobile strategy to employ, developing a native mobile application or a mobile website. Deciding on the mobile strategy depends on the needs of the planned mobile solution including targeted audience, intended purpose and required features.

Native mobile applications are developed for specific operating systems such as iOS or Android. The main advantage of native applications is that they benefit from the capabilities of mobile devices and are optimized to a specific platform. In order to use them they need to be downloaded from platform-specific portals or marketplaces (e.g. the Apple App Store or the Google Play Store) and installed on the mobile device. They are built with device-native programming languages. Unlike a native application, a mobile website does not need to be installed and can be accessed via the mobile browsers across all platforms. They are developed using typical web development technologies and languages such as HTML, CSS and JavaScript.

The previous phases of the design process provided key indications that developing a mobile website suits the scope of this thesis best. The main reason for a mobile-optimized website is that the mobile solution aims to complement the existing desktop version of Darwin. Therefore, the first step was to provide a mobile web solution before developing a native application. Furthermore, the development of a native application becomes very time-consuming when it is supposed to be available across multiple platforms. A mobile web application has the advantage that it is cross-platform and cross-device available. This also leads to lower maintenance effort in the future.

5.5.2. Mobile Frameworks

With the growing importance of mobile websites a large number of mobile frameworks evolved. They provide solutions which make mobile websites appear like native applications. They are also able to efficiently make use of device features such as the camera and touch input. Different frameworks have different strengths and weaknesses to create mobile optimized solutions. For this work two mobile frameworks were evaluated, Semantic UI and Angular Material. According to a set of predefined criteria they were compared with each other to find the one best suitable for the implementation.

The criteria were derived from [31] and are categorized into two groups: binary and qualitative. Binary criteria can be considered as questions regarding specific properties which can be answered with yes or no, e.g. platform compatibilities. Qualitative criteria address questions about quality properties of the framework and determine on what degree these properties can fulfill the specified requirements. Table 5.2 gives an overview of all the criteria including the assessment of each framework towards them. Each criterion is assigned a ranking from one plus (+) to a maximum of three pluses (+ + +) which scales as very good.

Binary criteria are a free software license and the possibility to integrate AngularJS into the framework. Both frameworks fulfill these criteria. They are both open source frameworks released under the MIT license. Angular Material and Semantic UI can both work together with AngularJS. However, Angular Material offers a large set of custom AngularJS directives and services, while Semantic UI's number of directives is limited to a small set. Functionality is triggered by JavaScript code snippets.

Criterion	Semantic UI	Angular Material
B1 Free license	✓	✓
B2 Integration of AngularJS	✓	✓
Q1 Mobile UI Support	+	+ +
Q2 Native Look and Feel	+ +	+ + +
Q3 Documentation	+	+
Q4 Ease of Learning	+	+ +

Table 5.2.: Overview of the framework assessment according to defined criteria.

Both frameworks offer various user interface elements such as forms, buttons and containers. Angular Material provides better support for mobile interfaces since it makes also use of touch gestures in its elements such as swiping and dragging. Angular Material also reaches a higher quality regarding the *Native Look and Feel*. It applies the same de-

sign found in Google applications which makes it recognizable to users. Another criteria is *Documentation and Support* which addresses support channels to learn the frameworks. Both frameworks are quite new. Angular Material has no official production release yet. However, both provide documentation with examples and community forums. The last criteria addresses the *Ease of Learning*. In general, the initial set up of an application is easy. Semantic UI extensively uses CSS classes which can lead to increased complexity for bigger projects and JavaScript code can become overwhelming. Angular Material applies the Model-View-Controller pattern which makes it easy to structure large amount of code. Based on the evaluation of the criteria it was found that Angular Material is best suited for the mobile web application developed for this thesis.

Part III.

Implementation and Evaluation

6. Technical Implementation

In this chapter the technical implementation of the mobile website is described in detail. Section 6.1 first gives an overview of the system architecture. Section 6.2 describes the technologies and frameworks that were used for the development of the system. Section 6.3 explains how the routing for the mobile website is realized. Section 6.4 introduces the data model and finally the Darwin API is presented in section 6.5.

6.1. System Architecture

The system is based on a client-server architecture with a presentation layer, a business logic layer and a data access layer. The conceptual architecture can be seen in Figure 6.1. This architecture model facilitates the separation of concerns increasing scalability, maintainability and performance of the system. The client-side comprises the **presentation layer** which is responsible for HTML templating. It provides the graphical user interface and enables the interaction between user and system. The server-side comprises the business logic layer and the data access layer.

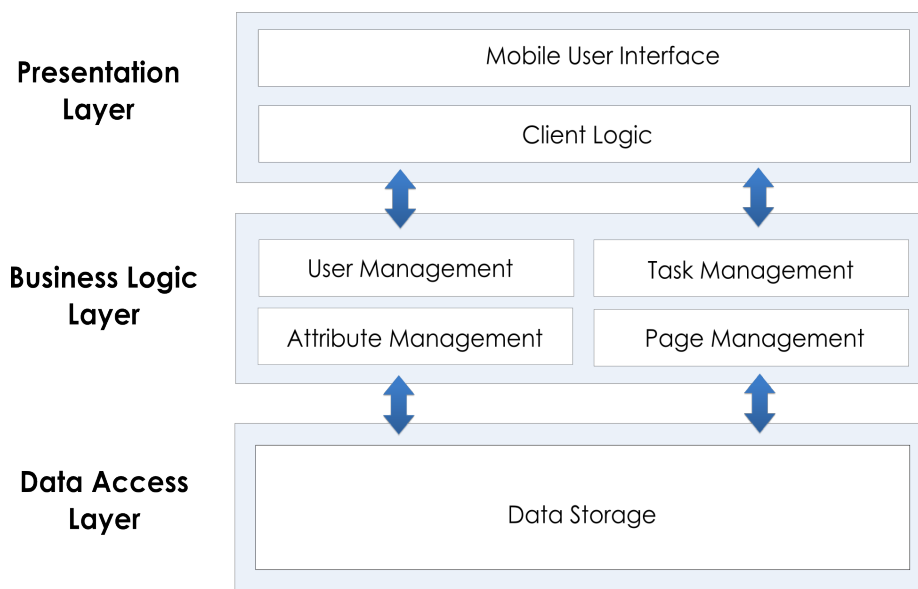


Figure 6.1.: System architecture.

All functionality is centralized in the second layer, the **business logic layer**. It provides management services and functionalities for users, wiki pages, tasks and attributes. The business logic can be shared by multiple front-end applications such as separated mobile and desktop clients. It further acts as interface to realize the request-response mechanism between client and server. The third layer is the **data access layer**. It holds the database management system and stores all the data of the application. In the following, a detailed explanation of the client and server-side architecture is provided, which is illustrated in Figure 6.2.

6.1.1. Client-Side Architecture

For the client side of the application the Model-View-Controller pattern is applied. This pattern separates the interface from the front-end application logic. The view displays the UI components and the data in the GUI. It comprises the HTML templates which are rendered in the mobile browser. The controller is responsible for the front-end application logic. It responds to user input and updates the model. It is further responsible for the communication to the server to exchange data by sending HTTP requests and receiving HTTP responses. The model is responsible for maintaining the data and serving it to the view on request.

6.1.2. Server-Side Architecture

The server-side logic is implemented with a routes file, models and controllers. The routes file represents the interface from the client to the server. It lists all request paths to controller methods with the corresponding standard HTTP methods, e.g. GET and POST. The client can then call the controller methods via the HTTP request paths. The controllers hold the core functionality and executes the specific methods after an HTTP request from the client. It retrieves, manipulates and stores data in the database via the models. The controllers also send data to the client in form of HTTP responses in the web format JSON. The model is responsible for the representation of the data on which the application operates.

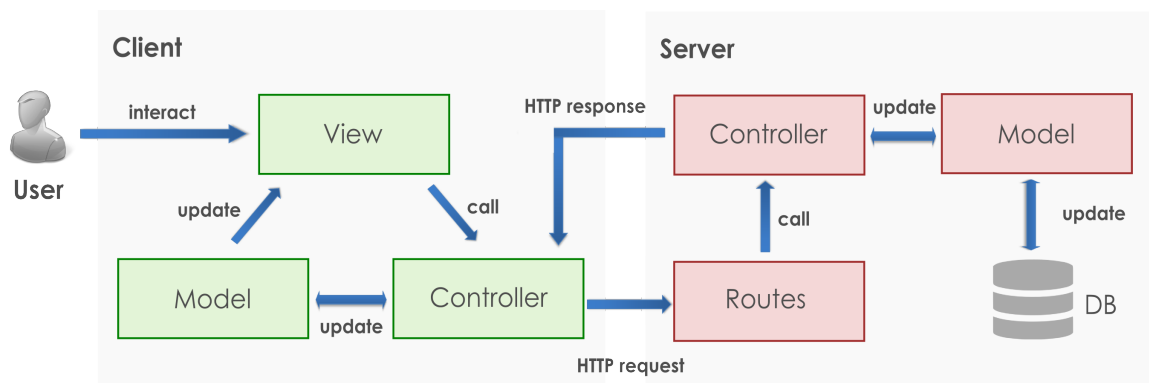


Figure 6.2.: Client and server architecture.

6.2. Technologies

For the development of the user interface Angular Material was used together with AngularJS. Angular Material is a responsive front-end framework with a set of AngularJS-native UI elements. It incorporates UI components with directives and services, allows custom theming and provides CSS for layout and styling. The framework is still in development process and available as pre-release. For the mobile website the version 0.10.1 was used. Angular Material realizes the concepts of Material Design in its elements. Material Design is a set of design guidelines introduced by Google in 2014. These guidelines specify how the interaction between users and devices should be designed through visual and motion interface elements. The idea behind the development of Material Design is to provide intuitive and natural interfaces by transferring attributes of real world elements such as shadows and textures to UI elements in the virtual world. Goals of these guidelines is to provide a comprehensive design language with principles of good user interface design and to create a unified user experience across all devices and platforms [25]. Angular Material supports developers during the design process by applying these guidelines to the UI components.

For the realization of the back-end components, comprehending the business logic and the data storage, the Play framework was used together with MongoDB as database. Play is an open-source framework which allows to build applications with Java and Scala. It is considered as a lightweight framework with a minimal resource consumption of CPU, memory and threads [32]. For the mobile application Play was used together with Scala. For the data access layer the MongoDB was used. MongoDB is a document-oriented database which stores the data in a JSON-like format in documents. The play controllers then can retrieve the data and send them as JSON files to the angular controller.

For the representation of several user interface elements additional libraries were included. For charts on the mobile website the JavaScript library Highcharts was applied. It provides a large number of different charts which are highly customizable. Angular Material does not yet offer a datepicker. Therefore, the angular-material-components library [64] was used to display a mobile-friendly datepicker. Moment.js is a library to parse, validate and manipulate dates in JavaScript, which was used in combination with the datepicker.

6.3. Routing

The mobile website was developed specifically for mobile devices and is accessible over the same URL as the desktop website. The system redirects mobile clients to the mobile site and desktop clients to the desktop site. Therefore, dynamic serving is applied to provide different HTML and CSS files depending on the access device. To realize correct routing a server-side method was implemented for the detection and analysis of the user agent. The browser always sends a user agent string as part of the HTTP request for identification to the web server. The user agent header contains tokens with information about the used browser, its version and the operating system [21].

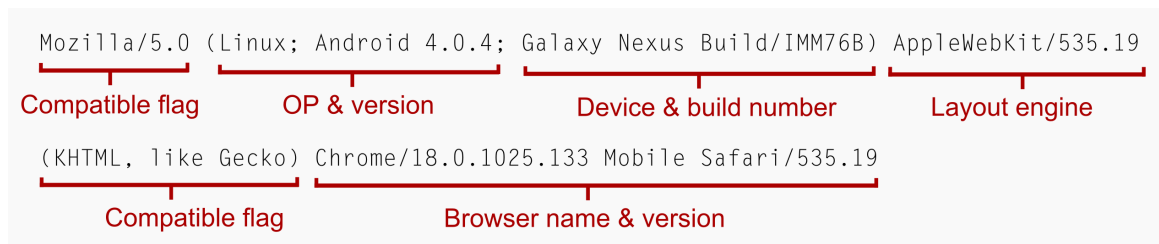


Figure 6.3.: Example of a user agent header.

Figure 6.3 shows an example of a user agent header and its tokens sent from a Galaxy Nexus phone using a chrome mobile browser. To identify the type of device the user agent header needs to be inspected for certain keywords. This is performed with a comprehensive regex in the `TemplateController` (see Figure 6.4). The regex contains keywords of devices and browsers which hint at mobile usage. The `isMobile` function inspects the user agent header and compares it with the regex. It returns true if the user agent header contains any of the keywords. In this case, mobile access is implied and the user is redirected to the index file of the mobile application. If the user accesses the site via desktop browser he will be directed to the appropriate templates of the desktop site.

```
object TemplateController extends Controller {

  val Pattern = "(iPhone|webOS|iPod|Android|BlackBerry|mobile|SAMSUNG|
    IEMobile|OperaMobi)".r.unanchored

  def index(any: String) = Action.async { request: Request[AnyContent] =>
    if(isMobile(request)) {
      Assets.at(path = "/public", file = "app/mobile/html/index.html").
        apply(request)
    } else {
      Assets.at(path = "/public", file = "app/html/index.html").apply(
        request)
    }
  }

  def isMobile[A](implicit request: RequestHeader): Boolean = { request
    request.headers.get("User-Agent").exists(agent => {
      agent match {
        case Pattern(a) => true
        case _ => false
      }
    })
  }
}
```

Figure 6.4.: Template controller for dynamic serving.

6.4. Data Model

In this section, a brief overview of the data model of Darwin Mobile is provided. It describes how the data of the system is collected and stored in the database. The target group of Darwin Mobile are non-expert end-users. Therefore, it has a more simplified data model than the desktop version which also provides data objects for the modeling of work templates. The simplified data model can be seen in Figure 6.5. It shows the main data objects of the system and their relations to each other.

The main data objects are Wikis, Wiki Pages, Tasks, Attributes and Users. A wiki can consist of multiple wiki pages. For each wiki page a type, tasks and attributes are defined. Wikis and wiki pages can also have informational text. Each attribute has a type and can have multiple values. A task can be specified with additional meta data such as start date, end date and task progress. The user can create wikis, wiki pages, tasks and attributes.

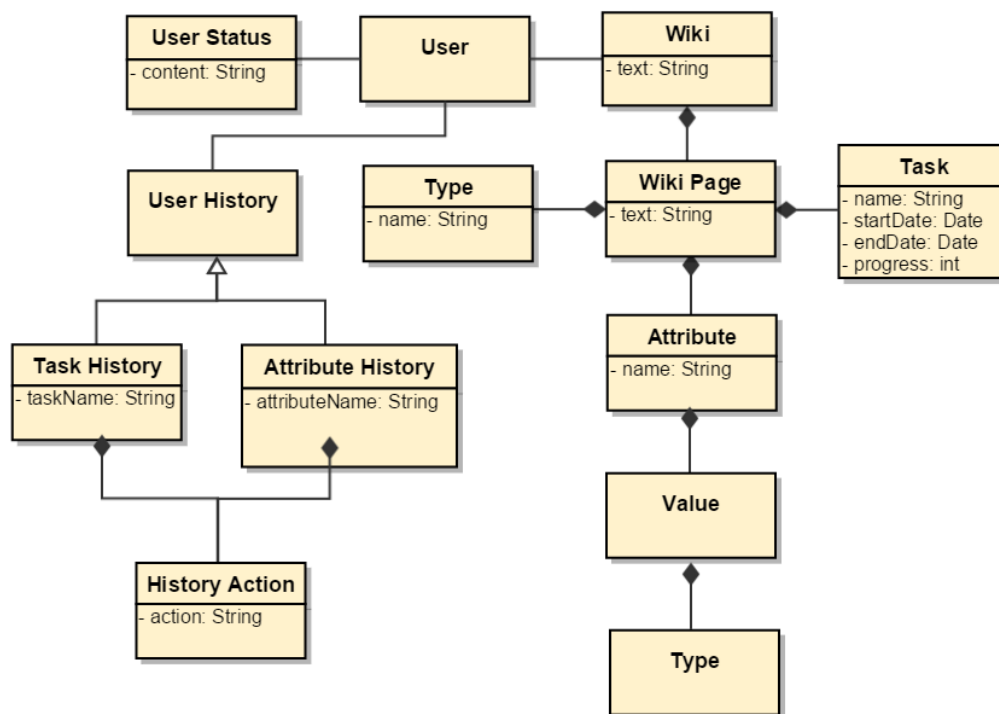


Figure 6.5.: Simplified class diagram of the datamodel.

The entries in the feed of the mobile website are categorized into the three groups tasks, discussion and data. Task entries comprise all activities around tasks, e.g. creating, updating or delegating tasks. Discussion entries are discussion posts of users on the feed and data entries are activities regarding attributes such as creating and deleting attributes or attribute values. To implement these three different categories two data models are defined, the `User History` model and the `User Status` model. Task activities are stored as `Task History` objects and data, respectively attribute activities, are stored as `Attribute History` objects. Both these objects are of the type `User History` and con-

tain a `History Action`. The `History Action` specifies the type of action, e.g. deleting, updating, creating. If a user creates a new discussion post it is stored as a `User Status` object.

6.5. Darwin API

The Darwin API makes method calls to different server-side controllers upon HTTP requests from the Angular controllers (see Figure 6.6). The user logs in and out of the system via the `AuthenticationController`. The `SearchController` performs the search through users, wikis and wiki pages. Via the `ModelController` wikis can be created and updated. The `PageController` is responsible for handling page text and attributes. The `ProcessController` is responsible for all interactions regarding tasks. The `UserController` handles the information about the user and provides the data and functionality of the feed. The `GroupController` is responsible for the management of the user groups. Users can be added and removed from groups as well.

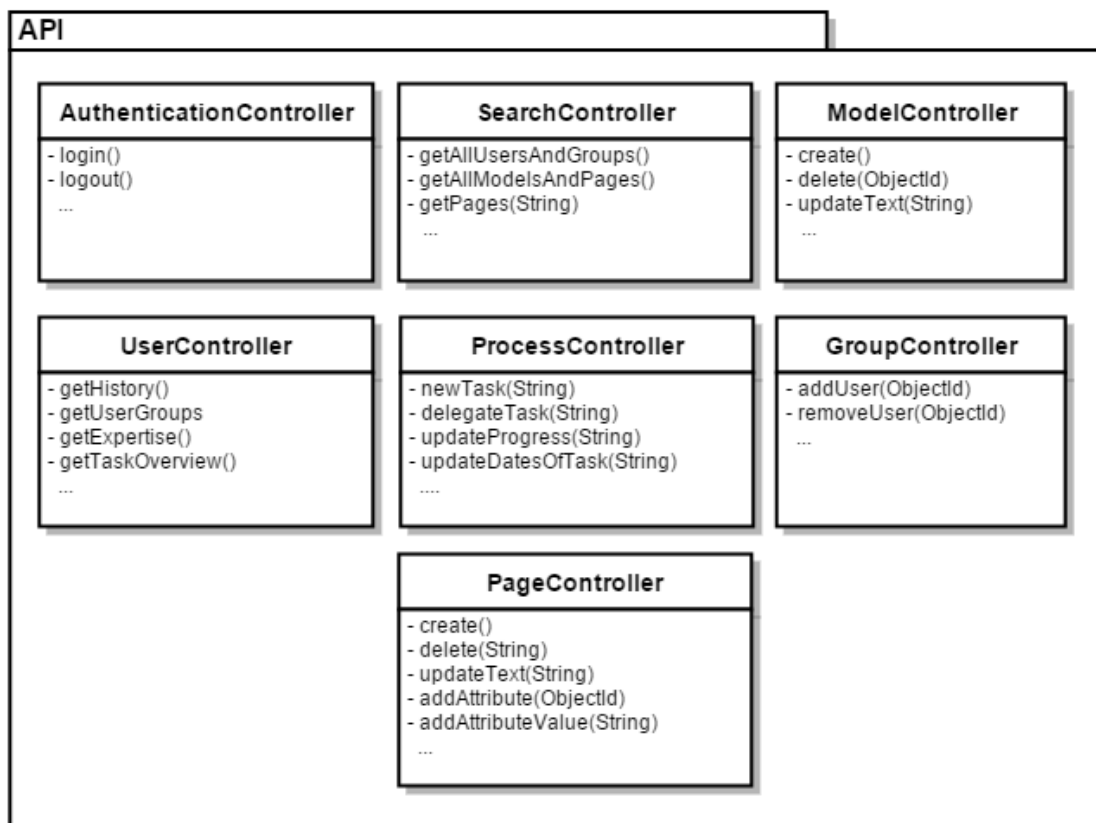


Figure 6.6.: Overview of the Darwin API and controller functions.

7. Design of the Mobile User Interface

The implementation of the front-end components of Darwin Mobile represents a significant part of this theses. This chapter provides an overview of the developed mobile user interface based on the requirements and task analysis conducted in chapter 5. Section 7.1 addresses the UI layout of the system including application structure, color, typography and theming. In section 7.2 the implementation of the UI components is presented with screenshots of the finished application.

7.1. Layout and Styling

7.1.1. Structure

The application consists of a permanent full-width toolbar on top, a content area below, a side navigation on the left and a drop-down menu on the right. The toolbar uses the hamburger icon as a control to open the side navigation drawer. This icon has found common use as menu button in many mobile applications. In the middle of the toolbar the current page title can be seen. The title lets the user know which page is currently displayed. On the right side of the toolbar app-related action buttons are placed. The vertical menu button provides access to the drop-down menu which contains links to user related pages such as the profile page. The search icon opens a search box which allows the user to search for wikis and wiki pages. Several pages use a tab structure to divide the content. The full structure of the mobile website can be seen in Figure 7.1.

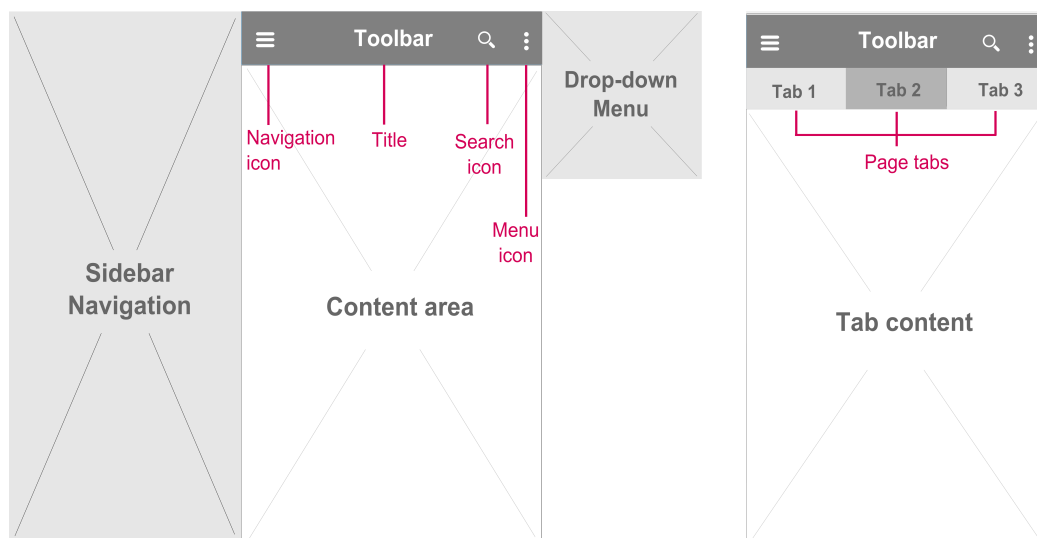


Figure 7.1.: Structure of the mobile website.

7.1.2. Color and Typography

When appropriately applied, color can have a powerful impact on design and interaction. It can enhance usability by creating visual hierarchy. Google Material has introduced 20 color palettes which work together harmoniously when combined. Each color has been assigned a specific number. The numbers are used for the custom configuration of a theme. The theming approach of Material Design advises to use three color hues from a primary palette and an accent color from a secondary palette. The primary palette is used for primary interface elements such as toolbars, links and buttons. It consists of a default color as well as a lighter and darker shade. Secondary interface elements are displayed with the accent color. The selected color schema for the developed mobile interface can be seen in Figure 7.2.

Primary - Light Blue		Accent - Cyan	
500	#3498DB	500	#00BCD4
200	#81D4FA	Warn - Pink	#F50057
500	#3498DB		
800	#0277BD		
		A400	#F50057

Figure 7.2.: Color schema of the mobile website.

For the primary colors, hues of the 'Light Blue' color palette were chosen. The default color was custom defined using the same blue shade which is also used on the desktop version of Darwin. The accent color was chosen from the 'Cyan' palette. Additionally, a warning color was defined for error messages using a hue from the 'Pink' palette.

Typography is another essential aspect of design. Just like colors, fonts can contribute to usability, an application's branding and most importantly reading comfort. Material Design's standard font is Roboto, a neo-grotesque sans-serif font family developed by Google. It is officially used for the Android operating system. Unlike many typefaces, Roboto offers six different font weights (thin, light, regular, medium, bold, black). To optimize the UI experience across multiple devices the font size is adjusted according to screen size, screen density and resolution. Angular Material uses predefined font sizes and font weights for different UI components.

7.1.3. Theming

Angular Material allows the configuration of an application theme with custom colors. This enables the user to apply the defined colors to different UI components by simply adding them as classes in the class attribute of the HTML or Angular Material tag. To configure a theme the `$mdThemingProvider` is used with a configuration method.

Figure 7.3 shows the specification of the theme with the color scheme presented in the previous section. Thereby, the default color of the 'Light Blue' palette was replaced with the custom blue color of the Darwin desktop application.

```
.config(function($mdThemingProvider) {  
  // extend and register color palette  
  var blueMap = $mdThemingProvider.extendPalette('light-blue', {  
    '500': '3498db'  
  });  
  $mdThemingProvider.definePalette('darwinBlue', blueMap);  
  
  $mdThemingProvider.theme('default')  
    .primaryPalette('darwinBlue', {  
      'default': '500',  
      'hue-1': '200',  
      'hue-2': '800' })  
    .accentPalette('cyan', {  
      'default': '500' })  
    .warnPalette('pink', {  
      'default': 'A400' })  
});
```

Figure 7.3.: Custom theming with Angular Material.

7.2. User Interface Components

In this section, the user interface components are presented. The user logs into the application over the login screen. After logging in with email and password the user is directed to the start page of the website which is the feed. The login page can be seen in Figure 7.4.

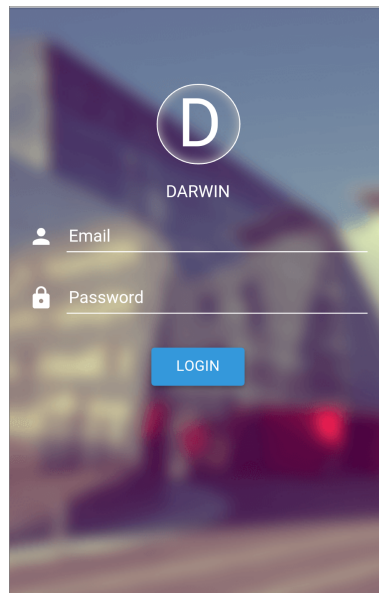


Figure 7.4.: Login page.

7.2.1. Navigation

The overall navigation of the website comprises a sidebar on the left and a drop-down menu on the right. All navigation options are available in the toolbar at all times to provide quick access for the user. The sidebar navigation applies the Side Drawer pattern. It is a common pattern which can be found in a large number of iOS and Android applications. The navigation slides from the left by clicking on the menu button with the hamburger icon. The user can close the side navigation by swiping to the left. The content of the side navigation is divided into three sections, the news section with access to the feed, the task overview section and the data section with the application's wikis. Wikis can be organized in a tree-like hierarchy. A wiki can have multiple wiki pages, which themselves can have subpages. To visualize this hierarchy of the wikis a sequential navigation structure is used. This allows users to easily traverse the hierarchy of the wiki data. A hierarchical navigation is generally applied for information-rich websites. Users click on menu items to reveal subordinate children menu items. An angle bracket icon on the right side of a menu item indicates that this menu item has child items. The first hierarchy level displays the wikis, the following hierarchy levels the wiki pages. Figure 7.5 depicts an exemplary navigation flow over three levels. When the user clicks on a wiki, e.g. *Web App Projects* the second navigation level is opened. On top of the second level a link is provided to the previous level so that the user can navigate back. Underneath, the current navigation position is shown, in this case the *Web App Project*. Finally, the user can continue to browse through the wiki pages which are displayed under the subpage category and if existing open subpages in the next sidebar level.

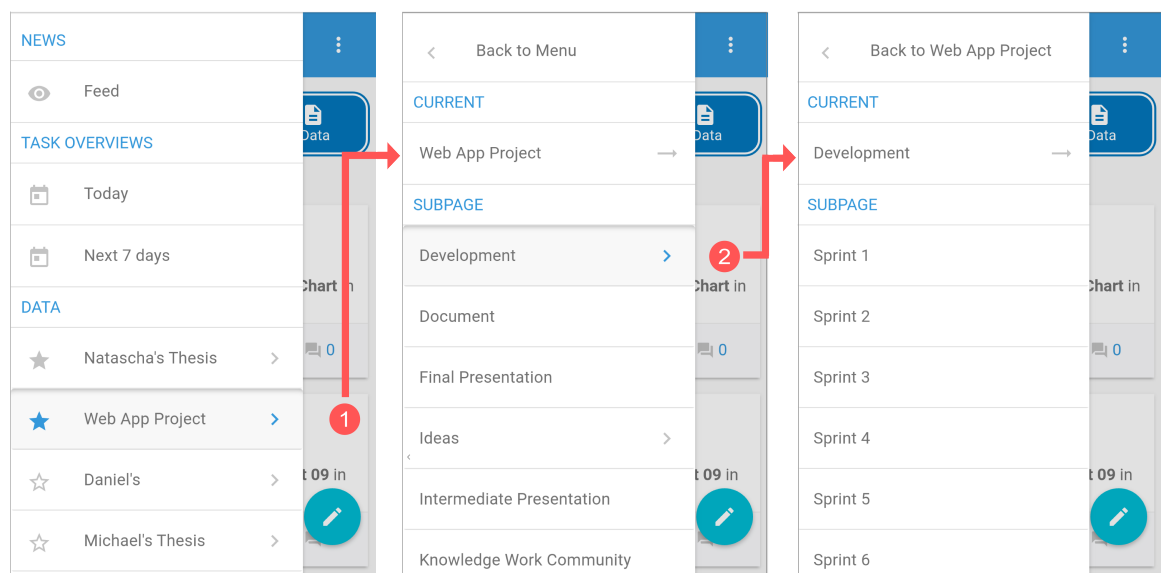


Figure 7.5.: Sidebar navigation drawer with three levels.

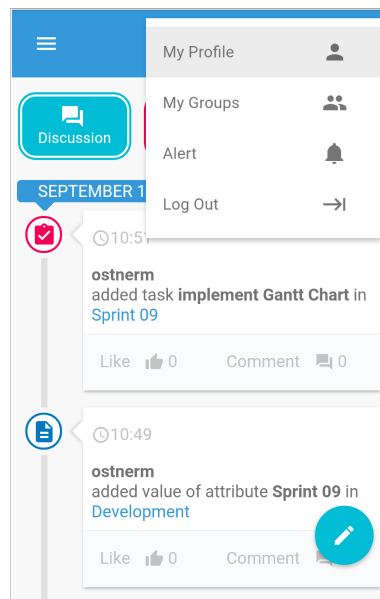
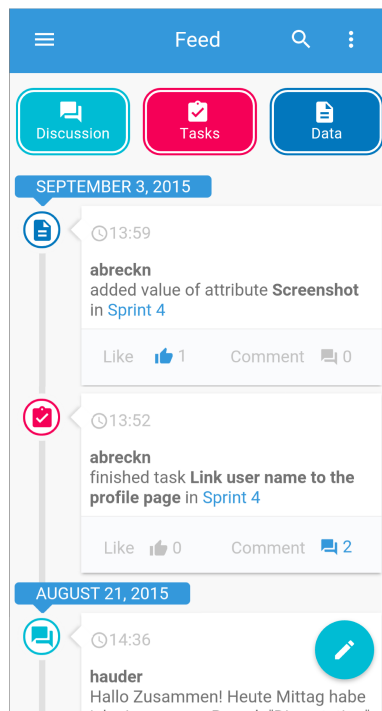


Figure 7.6.: Drop-down menu.

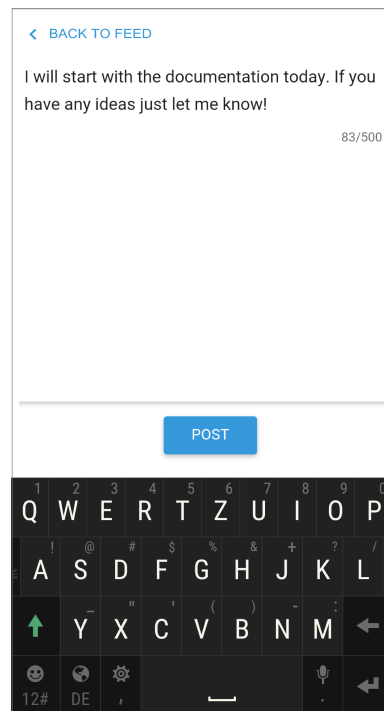
The right drop-down menu provides access to user-related content such as the profile page, the alert page and the groups page (see Figure 7.6). The user also can log off the website over the drop-down menu.

7.2.2. Activity Feed

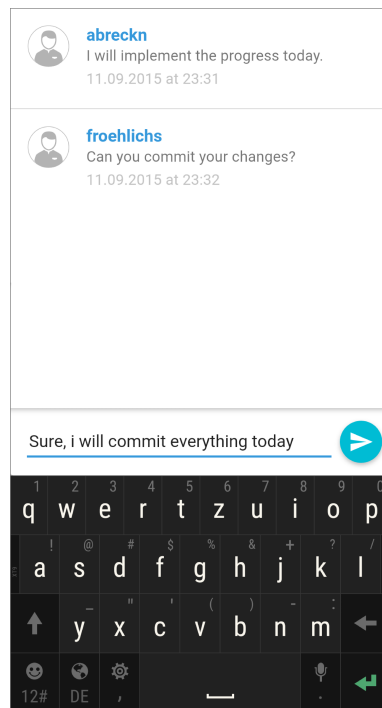
The feed gives an overview of the activities of all users in a timeline. Activities are grouped by date and sorted by time in a descending order. The feed is displayed in Figure 7.7(a). Each feed entry is assigned one of the following categories according to the user's activity: task, data or discussion. To indicate to which category an entry belongs, category-specific icons and colors are used. They are displayed on the left side of each entry. The number of loaded entries in the feed can increase rapidly due to extensive scrolling on the small screen. Users may then be overloaded with too much information afflicting the capacity of their cognitive load. Therefore, a filter on top of the feed is provided to enable users to efficiently refine the feed entries according to categories. Users can select multiple filters simultaneously to narrow down the feed entries. The feed implements further aspects of social media strategies. For example, users can comment and like user activities. Figure 7.7(b) shows the dialog for adding a new comment to an entry. The floating button on the right bottom of the page opens a dialog so users can create a new discussion post which can be seen in Figure 7.7(c).



(a) Timeline of the feed.



(b) Creating a new discussion post.



(c) Adding a new comment.

Figure 7.7.: Overview of feed components.

To continuously stream content on the feed page a lazy loading feature, also called *infinite scroll*, is implemented. Feed entries are automatically loaded when the scrollbar reaches a specific distance to the bottom of the page. Other solutions such as site paginations are not suitable for the mobile use. Also preloading large amount of data will leave users with increased page loading times. The general concept of the infinite scroll is illustrated in Figure 7.8.

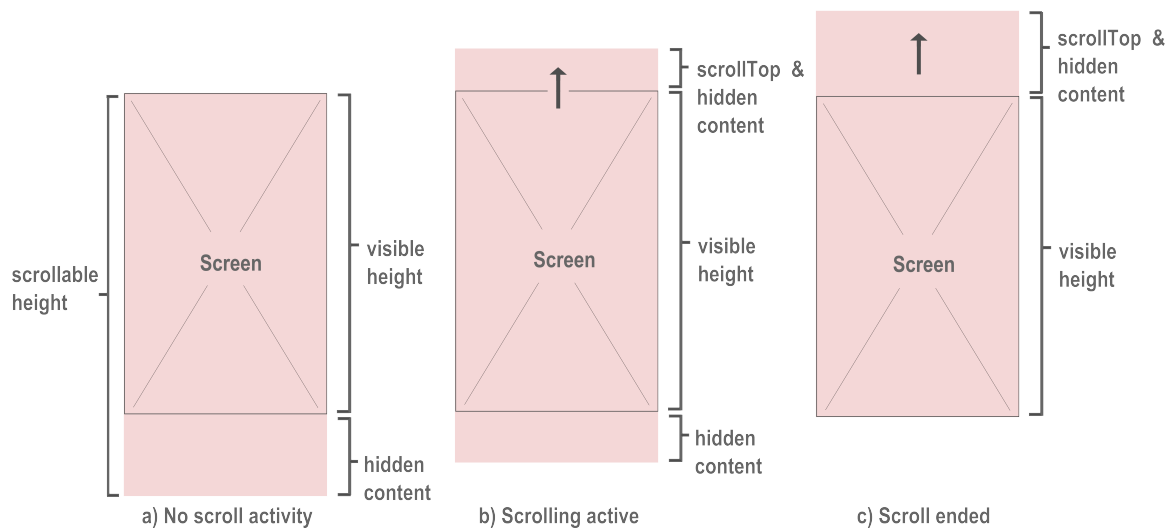


Figure 7.8.: Concept of infinite scroll.

The **visible height** represents the visible content. In this case, it is simply the screen size of the device in pixels. The **scrollable height** is the overall height of an element including the hidden content, which is not visible on the screen. The **hidden content** can be calculated by subtracting the visible height from the scrollable height. When the user starts scrolling, the position of the scrollbar changes. The position is calculated with the HTML DOM property **scrollTop**. **scrollTop** gets the number of pixels that are scrolled upwards. In Figure 7.8(a) **scrollTop** has a value of 0 because the user has not scrolled yet. When the user starts scrolling, the **scrollTop** value increases as seen in Figure 7.7(b). **scrollTop** reaches its maximum value when the user has scrolled to the bottom. This maximum value is equal to the hidden content height. Therefore, to determine if the scrollbar has reached a specific distance to the bottom of the page, the difference between the hiddenContent and the **scrollTop** is calculated. If the difference is smaller than the defined threshold to the bottom, new content is loaded. This increases the scrollable height again and the user can keep scrolling the page. This process is executed until there is no more data to load.

To implement this feature the custom directive `infiniteScroll` is defined. The directive watches for a scrolling event. When the user scrolls the event is triggered and the directive continuously calculates the difference between **scrollTop** and the hidden content. If the maximum threshold of 100px is reached a callback is executed invoking the `loadMoreEntries` function. Figure 7.9 shows the custom directive.

```
myDirectives.directive('infiniteScroll', function($window) {
  return {
    restrict: "A",
    link: function(scope, element, attrs) {
      element.bind('scroll', function () {

        var visibleHeight = element[0].offsetHeight;
        var scrollableHeight = element[0].scrollHeight;
        var hiddenContent = scrollableHeight - visibleHeight;

        if (hiddenContent - element[0].scrollTop <= 100 ) {
          scope.$apply(attrs.infiniteScroll);
        }
      });
    }
  };
});
```

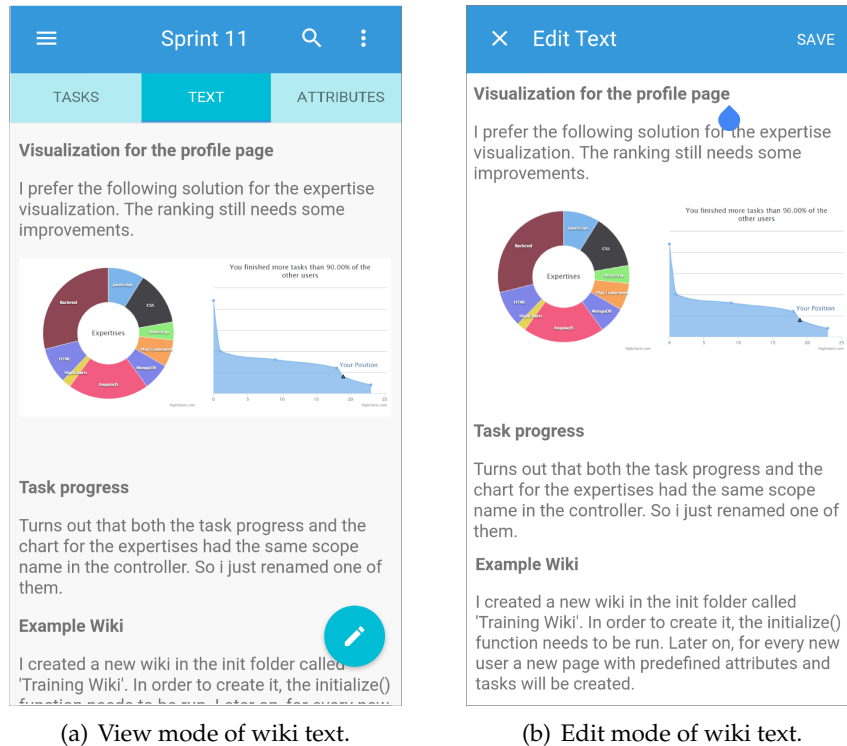
Figure 7.9.: Infinite scroll for the feed.

7.2.3. Wiki Pages

The content of wiki pages is structured in a tab format. Tasks, attributes and text are each displayed in a single tab. The currently selected tab is highlighted with a darker color. Users can switch between tabs by swiping.

Wiki Text

The middle tab contains the wiki text (see Figure 7.10(a)). Users can create and update text by clicking on the floating button with the edit icon. The desktop version of Darwin uses a text editor with an extensive toolbar to style text, add images or links. When the user for example applies a bold font weight to a word, this word will be saved inside an HTML `<b>`-tag. Loading the text in the view will then render the word together with the `<b>`-tag. In this way, all the formatting will be saved. If a user adds an image, it will be stored inside an `<img>`-tag with the link to the image. Due to the limited screen size, the use of a text editor is not feasible for the mobile website. Nevertheless, users should still be able to see the wiki text with all its styles and images when they edit it on the mobile device. Text inside standard HTML form controls such as input fields and textareas, however, cannot be styled. Therefore, to not lose the formatted text while editing, it is loaded into a regular `<div>`-tag, which is editable, instead of an HTML textarea. This will keep all the styles and images of the text. Figure 7.11 shows the HTML code and an AngularJS directive which were implemented.



(a) View mode of wiki text.

(b) Edit mode of wiki text.

Figure 7.10.: Text of wiki page.

The wiki text is rendered inside the `<div>`-tag by using the AngularJS `ng-bind-html` directive which is binding HTML to the view. The `contentEditable` attribute makes the text editable. Two way binding of the edited text inside the `<div>`-tag is necessary to save applied changes. Therefore, a custom form control as a directive was implemented. It retrieves the text value from the DOM and saves it in `$scope.editedText` in the hidden textarea. Figure 7.10(b) shows interface in edit mode including images and formatting.

```
HTML:
<div ng-bind-html="text" contentEditable></div>
<md-input-container style="display: none;">
  <textarea ng-model="editedText"></textarea>
</md-input-container>

AngularJS Directive:
myDirectives.directive('contenteditable', function() {
  return {
    restrict: 'A',
    link: function(scope, elm) {
      elm.bind('blur', function() {
        scope.editedText = elm.html(); });
    }
  };
});
```

Figure 7.11.: HTML code and Angular directive for editing wiki text.

Tasks

The left tab of the wiki page contains the tasks. Open tasks are displayed in a descending order according to the due dates of the tasks. Above the task description with name and due date, a colored bar indicates the progress of the task. Finished tasks are displayed beneath in a collapse panel. If the user clicks on the *completed tasks*-button the finished tasks for this page are shown.

Users can create and edit tasks for pages. To create a task, the user needs to click on the floating button with the plus icon. This will open a dialog for a new task. To edit a task, the user needs to first select the task. The icon of the floating button will then change to an edit icon. This will then open the edit dialog for the task. When creating or editing a task the user can provide several meta data, such as name, start and end date, task progress and expertises. Furthermore, a task can be delegated to one or more users or groups. For selecting a date, a custom AngularJS library was used since Angular Material does not yet offer any date picker. The progress of a task can be set with a slider. To assign a task to a user, an autocomplete functionality was implemented together with a search. The Angular controller loads all users and groups from the back-end and saves them in a scope. When the user starts typing, the scope is filtered accordingly and a list of suggestions is displayed. Selected delegates and expertises are displayed in Angular Material *chips*. The user can easily remove chips by clicking on the *cancel*-button next to each chip. The task overview and the edit dialog can be seen in Figure 7.12.

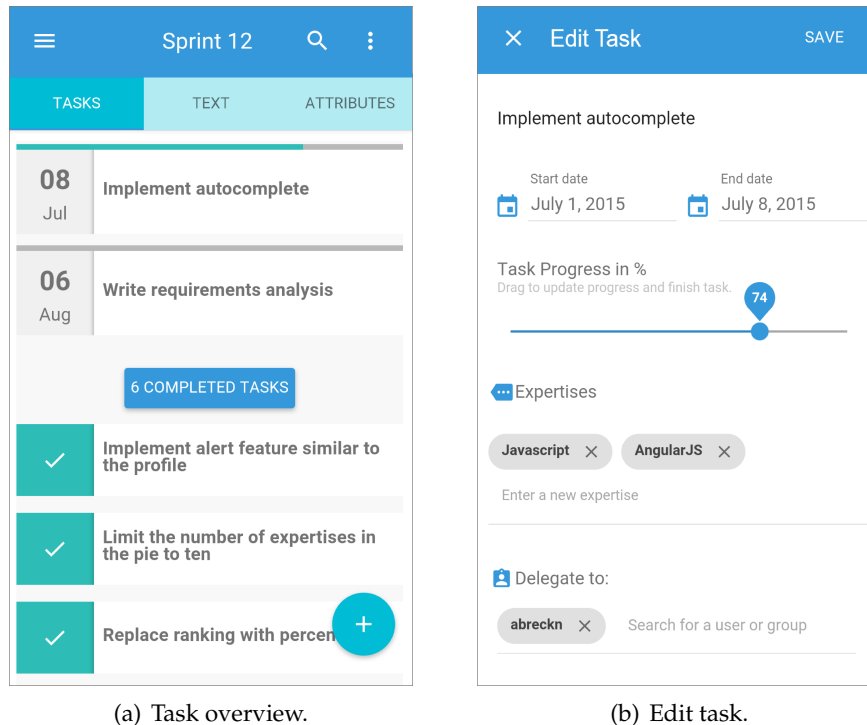
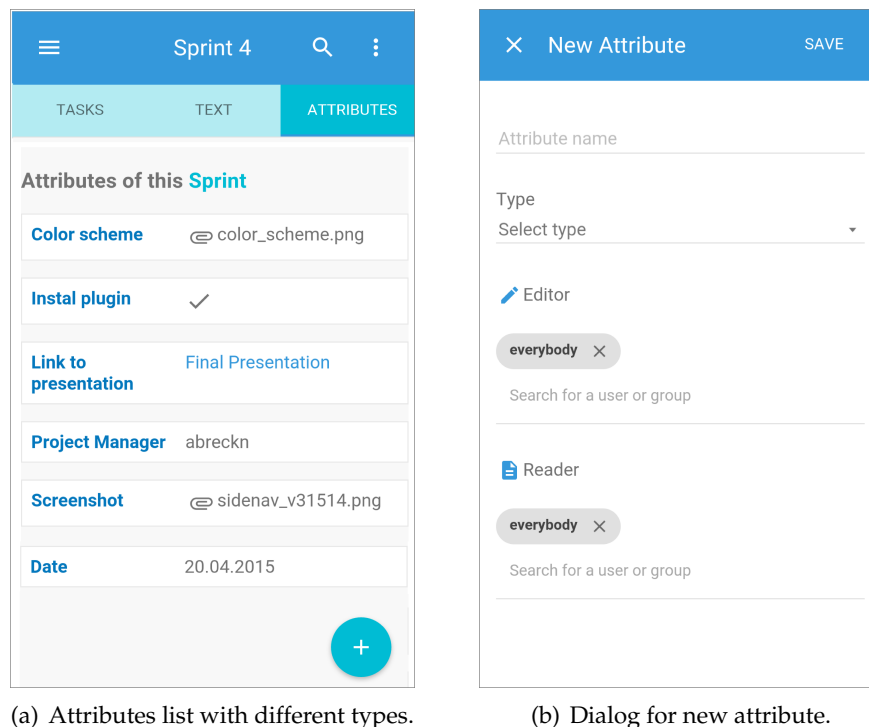


Figure 7.12.: Components of the task overview.

Attributes

Page attributes are found in the right tab. They are displayed in a list with the attribute name on the left and the attribute value on the right. When the user creates a new attribute he must select one of the seven attribute types. If the attribute is of type file a small file icon is set to the left of the attribute value. If the attribute is of type page, the attribute value is a link to a wiki page. Boolean values are represented with checkboxes. All other type values are displayed as simple strings.

Users can for now only create new attributes but not edit their values. They must define an attribute name and select an attribute type from a list. They can furthermore set read and write rights to attributes. Initially, read and write rights are assigned to all users. When a user selects a task from the task view, the attributes will be filtered according to the selected task. Then only mandatory attributes for the task will be displayed. Figure 7.13(a) shows an overview of page attributes of different types and Figure 7.12(b) shows the dialog for creating a new attribute.



(a) Attributes list with different types.

(b) Dialog for new attribute.

Figure 7.13.: Components of the attribute overview.

Besides editing existing content, the user can also create new objects such as wikis or wiki pages. The user can open the dialog for the creation of new objects over the side navigation. Figure 7.14 shows the dialog to create a new wiki page. The user needs to select a type of object. If the type *wiki* is selected the user must only define a wiki name. If the type *page* is selected the user is presented with additional fields. He must define a name, a page type and the superior page. He must define a name, a page type and the superior page.

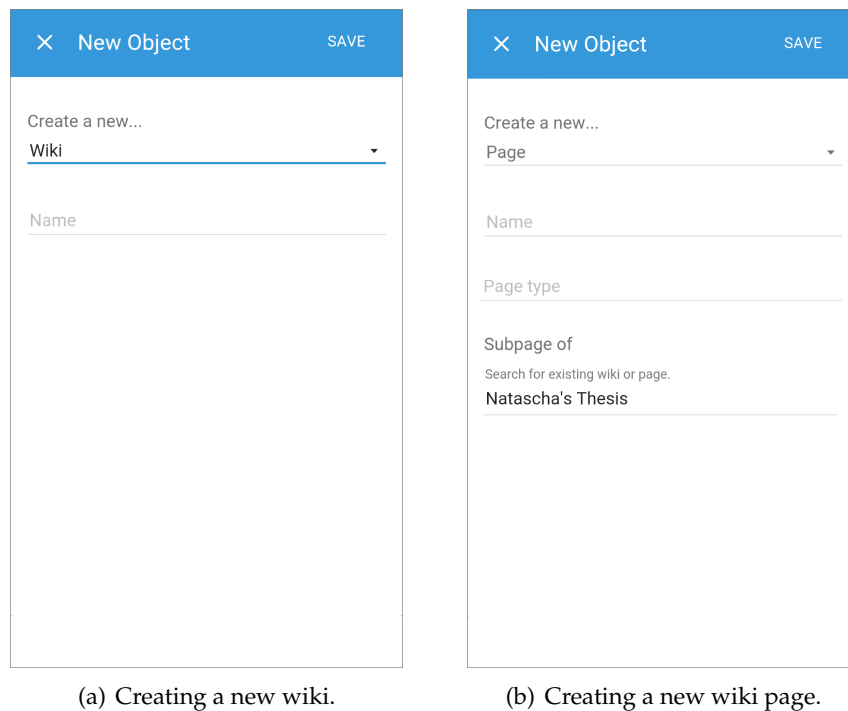


Figure 7.14.: Creating new objects.

7.2.4. Profile Page

The profile page also applies a tab view to structure its content. The components of the profile are closed tasks, open tasks and user expertises. On top of the profile page the profile picture is displayed. Around the picture a chart can be seen. This chart represents the percentile ranking of the user. The percentile ranking calculates how many tasks the user has completed compared to all other users. A percentile ranking of e.g. 90 indicates that the user has completed more tasks than 90% of the other users.

Below the profile picture the tabs are displayed. The middle tab gives an overview of all open tasks of the user. The left tab contains all the tasks the user has finished and the right tab contains the expertises of the user. A user can collect expertises if he finishes a task which is tagged with specific expertises. The expertise overview also shows how many tasks the user has finished for each expertise. A maximum of six different expertises is displayed. These are the expertises with the most accomplished tasks. The remaining expertises are grouped as *other* expertises. The Highcharts library was used to visualize

the expertises and the percentile ranking. The expertises are displayed with bar charts, the percentile ranking with a pie chart. Figure 7.15 shows an overview of the profile page.

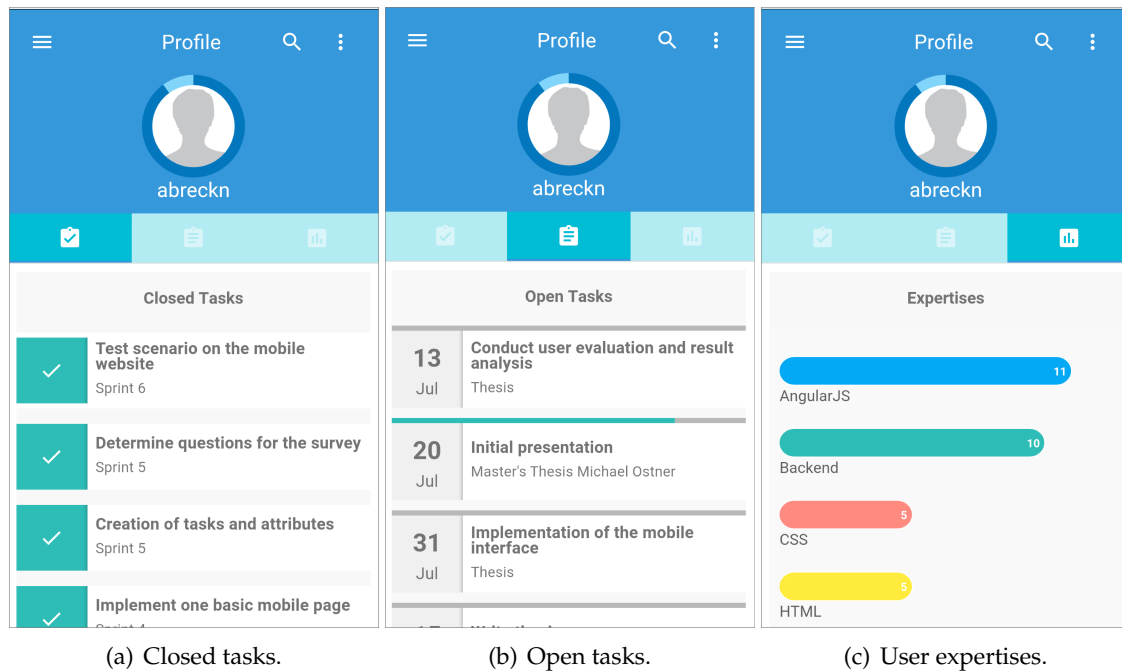
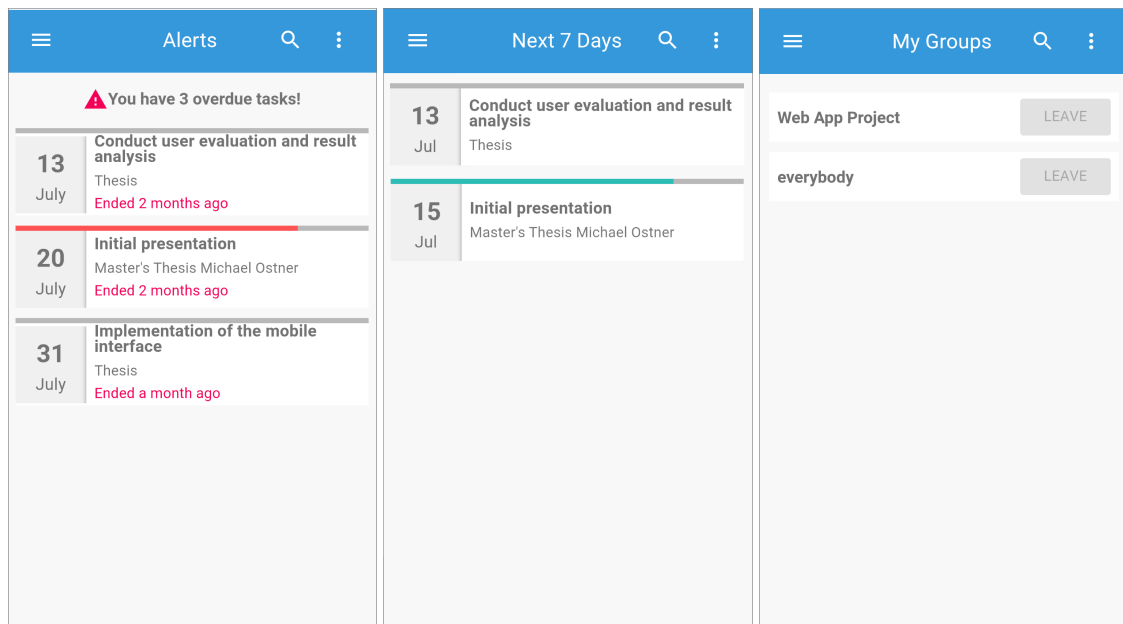


Figure 7.15.: Components of the profile page.

7.2.5. Secondary Pages

The mobile website contains a number of additional pages which are briefly described in this section. Reminders are popular features of existing task applications. Therefore, an alert page is implemented which gives the user an overview of all overdue tasks (see Figure 7.16(a)). It can be accessed over the drop-down menu in the toolbar. The alert shows the task including task name, wiki name, due date and the duration of the expired time. Two other additional task overviews are provided for the user which also have a reminder functionality. The overview of the tasks of the current day and of the following seven days. Figure 7.16(b) shows the open tasks for the next seven days. The groups page gives an overview of the groups the user is a member of. However, only users with administrative rights can leave or join groups. The groups page is displayed in Figure 7.16(c).



(a) Alert page.

(b) Task overview of the next seven days.

(c) User groups page.

Figure 7.16.: Overview of secondary pages.

8. User Evaluation

Evaluations represent an important part of the development life cycle. The early assessment of a system helps to identify issues during the implementation phase and to apply necessary changes in an iterative design process. Usability evaluations thereby aim to measure user satisfaction, performance and how well users can achieve their goals. A user evaluation of the developed mobile website was conducted to determine how well users could execute knowledge intensive tasks with the mobile solution and how they perceived the user interface from a usability point of view. In this chapter, the evaluation of the implementation and its results are presented. Section 8.1 begins with the introduction of common usability evaluation methods. Section 8.2 presents the evaluation methodology, the results and the discussion, which documents the implementation of the evaluation's findings into the existing solution.

8.1. Usability Evaluation Methods

A usability evaluation method is a procedure which consists of a set of defined activities to gather usage data. This usage data gives indications about the end-user interaction with the evaluated system. Several taxonomies of usability evaluation methods have been proposed in the literature. A common taxonomy classifies evaluation methods into two groups, empirical methods and inspection methods [10, 18, 46].

Empirical methods are used to collect and analyze usage data from real end-users. Thereby, end-users execute predefined tasks while evaluators collect information by observing them. The users can provide useful opinions and suggestions about the system in a real work environment which help to uncover and detect usability problems [52]. Empirical methods require a large amount of resources, e.g. laboratory equipment, and are often time consuming and expensive. Furthermore, to conduct an empirical evaluation a functional implementation must be available. Frequently applied empirical methods are the *Think-Aloud Protocol* and *Remote Usability Testing*, which are explained in more detail below.

- **Think-Aloud Protocol:** During the Think-Aloud Protocol participants state their personal feelings and impressions about the system while they are performing given tasks. This method allows the evaluator to directly exhibit where users might have problems or where misconceptions occur due to the interface design. The only challenge of this method is that it might interfere with the participant's natural usage of the system due to the strict observation. However, if the evaluator does not interrupt the user with questions or comments during the session, this method will not interfere with the task performance of the user [16].

- **Remote Usability Testing:** During a remote usability test evaluators and test participants are geographically separated. Audio and video communication allows to observe the user while he is performing the given tasks [29]. Automated remote usability testing does not require the presence of the evaluator during the test. User data is collected automatically by observation systems and stored for later analysis [65].

Usability inspection methods are performed by usability specialists and expert evaluators. These evaluation methods do not require the participation of real end-users. They are rather based on the expertise and experience of the evaluators. Thereby, the usability aspects of the system interface are examined with consideration of usability standards, guidelines or a set of predefined criteria [46]. Inspection methods usually fit naturally in the development life cycle making them easy to conduct even in early stages of the development. For usability inspections a functional implementation is not essential, since they can be conducted with other artifacts such as mockups or paper prototypes. These characteristics make them very cost-effective. [46]. A short overview of the most common techniques is provided in the following. [46].

- **Heuristic evaluation:** Usability experts evaluate if an interface follows established usability heuristics.
- **Cognitive walkthrough:** This is a task-oriented method for which a usability expert simulates typical user behavior. The tasks for the walkthrough are predefined. The evaluator inspects system functionalities and examines if user goals can be achieved correctly.
- **Pluralistic walkthrough:** This is a version of the cognitive walkthrough which includes UI designers, developers and users in addition to the usability experts. In regular meetings, the evaluators go through typical user scenarios and accordingly evaluate the UI.
- **Feature inspection:** For this inspection method, use cases with specific results are defined. Evaluators perform these use cases on the user interface and test single features regarding task completion, understandability, ease of learning and memorability.

8.2. Empirical Evaluation of Darwin Mobile

For this work, a user evaluation of the mobile web application was conducted. The overall objective of the evaluation was to early identify usability problems of the mobile solution. The evaluation was conducted midway into the development cycle so that there was enough time left to provide suitable solutions and implement necessary changes into the system.

8.2.1. Test Objectives

The user evaluation aims to examine the usefulness of Darwin Mobile to structure knowledge-intensive processes in practice. It further helps to gather baseline data about the effectiveness of the system. Overall objective of the evaluation is to answer the following research question: "How well can the mobile interface be used to accomplish a knowledge-intensive task in practice?" The results of the evaluation will help to discover possible hurdles regarding the management of knowledge-intensive tasks in a mobile environment and identify usability problems of the mobile interface. The research question is divided into following sub-questions:

- **Question Q1:** How well do the end-users experience the mobile interface?
- **Question Q2:** How effectively can end-users work collaboratively with the mobile solution?
- **Question Q3:** How easily can end-users manage knowledge-intensive tasks on a mobile device?
- **Question Q4:** How can the design of the mobile user interface be improved based on a user-centered design approach?

8.2.2. Methodology

To assess the usability of the mobile website an empirical evaluation method is applied. The participants are given predefined tasks which must represent typical user actions. The users will perform the given tasks commenting upon their actions, impressions and thoughts about the system. The Think-Aloud Protocol suggests little or no interaction with the participants. Hence, participants are executing the tasks autonomously, but are given assistance if asked for support.

After users complete their tasks, they will fill out a questionnaire about the system. This inquiry method gathers additional subjective data from participants. For the questionnaire, the *System Usability Scale (SUS)* is applied [6]. SUS is a ten-item questionnaire with five response options for the subjective assessment of usability. Figure 8.1 shows the complete SUS questionnaire which was used for the evaluation.

SUS uses the *Likert* scale as a response format. Thereby, participants answer in terms of the extent to which they agree or disagree with a statement, ranging from "I strongly agree" to "I strongly disagree" in a five point rating scale. SUS applies positive and negative statements to measure frequency of use, complexity, learnability and ease of use. The alternation of positive and negative statement items reduces response bias (participants providing all high or all low ratings). SUS is considered as highly reliable, meaning that it will deliver the same outcome if repeated [7]. To gather more specific information about the participants' impressions of the system, they were given the opportunity to leave additional feedback in comment fields next to each questionnaire item.

	Strongly Disagree				Strongly Agree
1. I think that I would like to use this product frequently.	1	2	3	4	5
2. I found the product unnecessarily complex.	1	2	3	4	5
3. I thought the product was easy to use.	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this product.	1	2	3	4	5
5. I found the various functions in the product were well integrated.	1	2	3	4	5
6. I thought there was too much inconsistency in this product.	1	2	3	4	5
7. I imagine that most people would learn to use this product very quickly.	1	2	3	4	5
8. I found the product very awkward to use.	1	2	3	4	5
9. I felt very confident using the product.	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this product.	1	2	3	4	5

Figure 8.1.: The SUS questionnaire [6].

8.2.3. Tasks

Five tasks were defined for the user evaluation, encompassing a complete test scenario. The tasks cover most frequent use cases of the application and were defined together with an expert user of the system. Table 8.1 shows the tasks, their test goals and the completion criteria.

Task Description	Test Objective	Completion Criterion
1. Write a comment to the last feed entry of your supervisor.	The user can apply the filter to find the specific feed entry and leave a comment.	The user comments on the right feed entry.
2. Go to the page of exercise 4 inside your project.	The user can navigate through the system.	The user navigates to the exercise 4 of his project.
3. Create a new task. Set start and end date of the task and add two expertises.	The user can create a new task and define meta data.	The user creates a new task with the specified meta data.
4. Go to the task overview for next week.	The user can navigate through the system.	The user opens the right overview.
5. Finish the task you created in step 3.	The user can finish a task by setting the progress to 100%.	The user finishes his task.

Table 8.1.: Task description of the user evaluation.

8.2.4. Participants

69 people participated in the user evaluation. All participants were students from a web application engineering class. In the course, students developed a web application in teams applying the concepts of the lecture. The students were instructed to use the desktop version of Darwin during the semester to retrieve exercise sheets and hand in exercise deliverables for the course. Every team had its own wiki page in Darwin to describe their project. Further, for every team the tasks of the exercises were transferred to tasks in the Darwin system. All tasks had file attributes assigned to them, which the students needed to fill in with deliverables in order to complete the tasks. Every student was briefed to at least submit one deliverable via Darwin. Hence, users were familiar with defining attribute values and uploading files to the system. They also used the activity feed to see updates or news about the course or to contact supervisors. However, they did not define new tasks on their own and hence, were not familiar with this feature.

8.2.5. Procedure

The evaluation took place at the Technical University of Munich. Since the mobile website was only partly developed and consistency through all mobile browsers could not be

assured, participants used a PC to perform their tasks. This also eased the process of observing the students. Through an internet browser, which was adjusted to a mobile view, the participants accessed the website.

The evaluation was conducted on two days in several sessions. In each session, one project team with up to four participants evaluated the mobile website at the same time. Prior to the evaluation the participants were greeted, briefly informed about the background and purpose of the evaluation and given instructions about the evaluation process. Participants were informed that they will be observed and were hence asked to express their thoughts out loud during the task execution. They were further informed that the system is in a preliminary stage but that all necessary features to perform the tasks are implemented. After the task completion participants filled out the questionnaire with the LimeSurvey tool.

8.2.6. Results and Analysis

This section presents the results and the analysis of the user evaluation. The results are split into two parts. First, the results of the usability test and the impressions from the Think-Aloud Protocol are presented. The second part summarizes the results derived from the questionnaire and the calculated SUS score.

Experimental Observations of the Think-Aloud Protocol

To perform the tasks the students required between five and ten minutes. All students were able to complete the given tasks successfully. Some of the participants required assistance to perform task four.

Task Nr. 1: Commenting on a feed entry

Most of the participants were not familiar with the filter option on top of the feed and did not use it to find the correct feed entry. Some participants who used the filter option commented that they did not find the selection of a filter consistent. Meaning, that they were not sure if clicking on a filter button selected or deselected the filter. Some stated that the filter buttons looked like tabs rather than buttons and hence prevented them to recognize them. The like and comment buttons were initially implemented in an accordion view. Some users stated that they found this unnecessary complex since it resulted in an extra click action for them. Few participants initially opened the dialog for a new discussion entry because the first thing they noticed was the floating button on the bottom of the display. However, all these participants corrected their action and were able to complete the task successfully.

Task Nr. 2: Navigation to the exercise page

All participants recognized the hamburger icon as the menu icon. Participants stated that they liked the design of the multi-level navigation. However, few students said that they

were overwhelmed by the amount of navigation items. This was due to the fact that overall 20 wikis were created in Darwin for the course, one for each team, and all these wikis were listed in the menu. Participants wished for a more personalized view which did not display the wikis of the other teams or at least highlighted their own projects.

Task Nr. 3: Creation of a task

Few of the students did not see the navigation tabs on top of the wiki page. Hence, some of them needed some time to navigate to the task view. Some students suggested to better highlight the "closed task"-button at the bottom of the task view. It was not conceived as clickable by all users. All users were able to open the dialog to create a new task by clicking on the floating button. Some stated about the task dialog that they expected the save button to be at the bottom of the page as seen in many web forms rather than on top in the toolbar. Few participants saved a task before they set the specified task meta data. However, after pointing it out to those users, all were able to edit the task and add the missing meta data afterwards.

Task Nr. 4: Opening the task overview

Almost all students had trouble navigating to the task overview. They did not know that they could navigate to the overview via the left navigation sidebar. Although some students intuitively did open the menu, they initially overlooked the navigation item for the overview.

Task Nr. 5: Finishing the task

Participants chose different ways to navigate back to the exercise page to finish their task. Some students directly navigated to the exercise page by clicking on the respective task visible in the open task overview. However, they stated that they expected the edit dialog to open directly when they clicked on the task. Other participants used the sidebar menu which resulted in more effort to navigate to the exercise page.

On the wiki page, when students clicked on the task they wanted to finish, they expected the edit dialog to directly open. However, on the mobile website, similar to the desktop application, the user first needs to select a task and then click the edit button to open the edit dialog. The selection of a task is necessary so the user can switch to the attribute view of the wiki page and see the attributes assigned to the selected task. The majority of the students found this complex and not intuitive. This was mostly due to the fact that students were not aware, that attributes could be assigned to tasks. They seemed more understanding when it was explained to them. Regarding the edit dialog, many participants were looking for a finish-task button or a checkbox to finish the task. Many students did not initially recognize the slider for setting the task progress. The pop-up which appeared after users created or finished a task was perceived very well. Students stated that it provided helpful feedback if their action was successful or not.

Results of the Questionnaire

The results of the questionnaire are presented in the following. They can be divided into two data sets: data regarding the learnability of the system and data about the usability. The usability data covers the aspects frequency of use, ease of use and consistency.

Frequency of use:

57% of the participants stated that they would not necessarily use the system frequently during the course if the mobile website was available. 23 participants additionally commented in the questionnaire that they used the platform only to hand in their deliverables but not to collaborate with their team members or to manage their project. Various participants also mentioned this during the usability test. Participants quite often referred to the website as the *submission system* indicating that they used it foremost to upload their files but not for management purposes. They further stated that a file upload via the mobile website is probably more complex than via the computer. According to the results, 26% would additionally use the mobile version. The results give indication that participants did not appear to be fully aware of the available features of Darwin, but rather reduced its functionality to uploading files. An overview of the results can be seen in Figure 8.2.

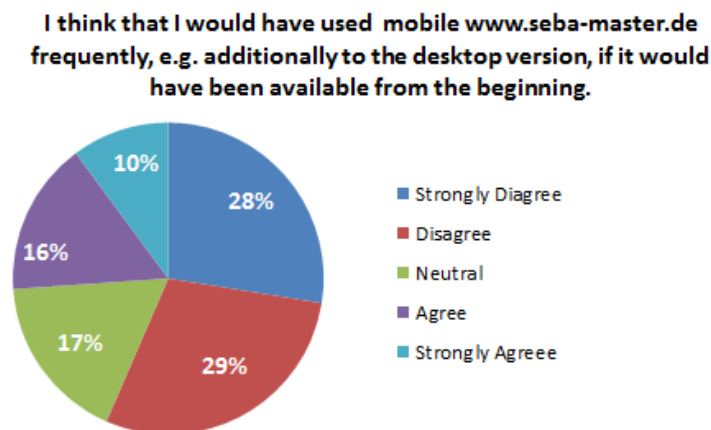


Figure 8.2.: Results regarding the frequency of use.

Ease of use:

The users were asked four questions regarding the ease of use of the system. Two of them were formulated as positive statements, the other two as negative statements. The results can be seen in Figure 8.3.

Regarding the question if the system was unnecessarily complex, 61% of the participants disagreed or strongly disagreed. However, 20% agreed or strongly agreed that the system was unnecessarily complex (see Figure 8.3(a)).

64% stated that they found the system intuitive and easy to use (see Figure 8.3(b)). 14% stated that it was difficult to use. Several participants provided additional feedback in form of comments to reason their rating. As mentioned during the usability test, some participants commented that they had trouble navigating through the system. This was due to the large number of wikis displayed in the side navigation. Many students stated that they wished a more personalized view of the navigation, e.g. instead of scrolling through all wikis, only their own wiki is displayed. Users further commented that the main challenge for them was to finish a task. They did not know that the progress of the task needed to be set to 100% in order to finish it. They also said that they never created tasks on the desktop application. 57% stated that they felt very confident using the system as novice users, 15% did not feel confident and 28% rated their confidence as average as seen in Figure 8.3(d).

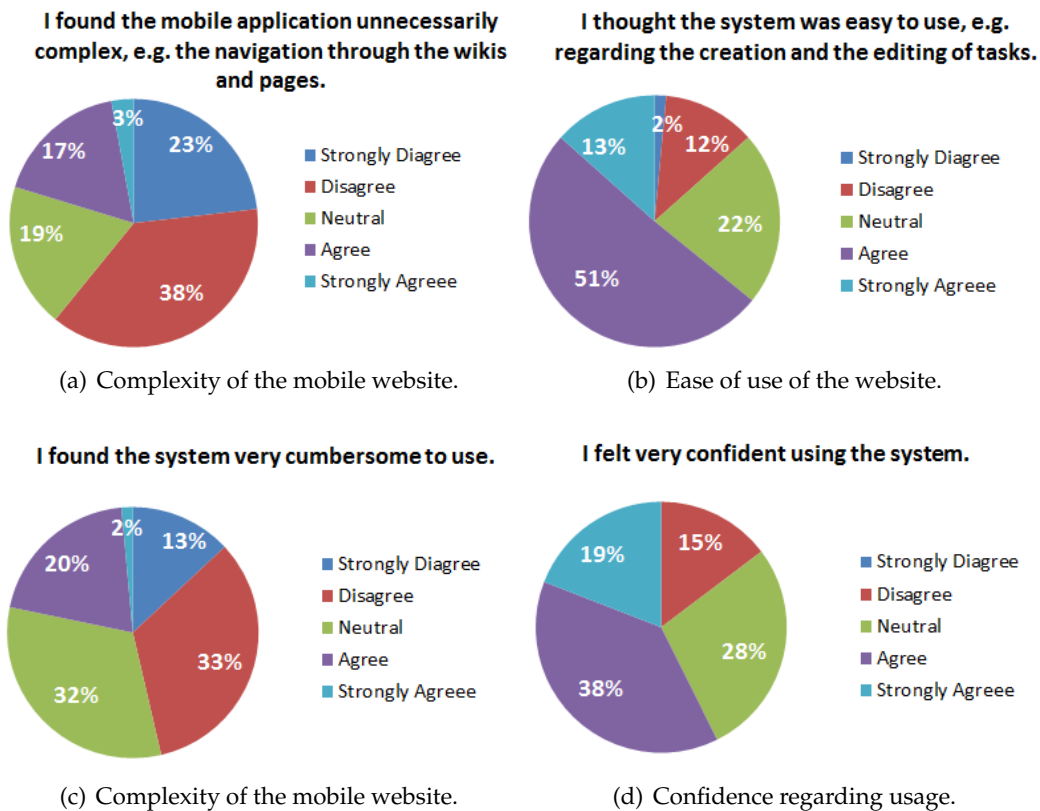


Figure 8.3.: Results regarding the ease of use of the website.

Consistency

Consistency addresses two aspects: the overall perceived consistency regarding the user interface design including colors, visuals such as icons or buttons, typography and structure of the content. The second aspect addresses the consistency between platforms, in this case between the desktop version of Darwin and the developed mobile version.

83% agreed and strongly agreed that the functionalities of the desktop website were well integrated into the mobile system (see Figure 8.4(a)). Only one participant, representing 2%, did not agree regarding consistency between the two systems. Altogether, 72% stated that they found the overall design throughout the website consistent (see Figure 8.4(b)). 23% of the participants had a neutral opinion about the consistency of the UI design, while only 5% did not agree.

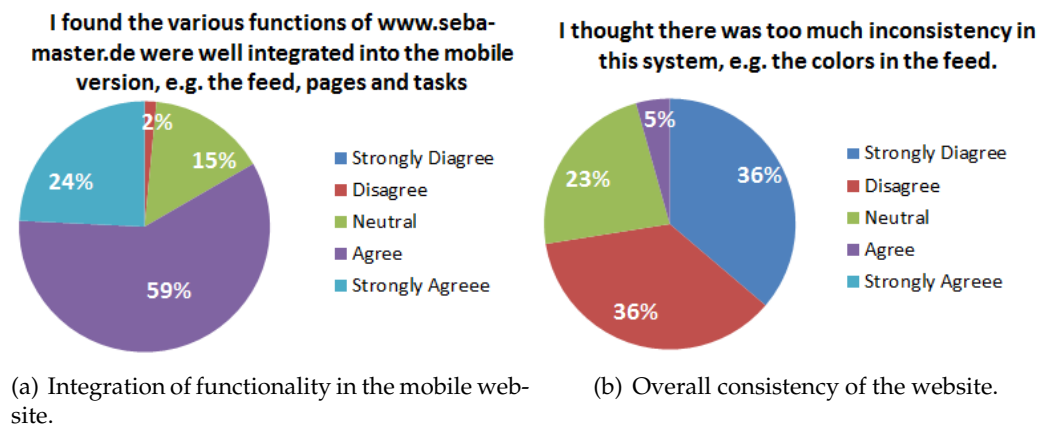


Figure 8.4.: Results regarding the consistency of the website.

Learnability

Questions four, seven and ten of the SUS questionnaire measure the learnability of the system. Figure 8.5 gives an overview of the percental distribution of the answers to the three items. The answers are consistent, revealing that the majority of the participants would learn the system quickly. This indicates that students experienced an easily learnable system and did not stumble across difficulties on their first use. Altogether, 72% of the participants answered that they did not require to learn the system or gain experience prior to using it. One participant said that he would need to learn a lot about the features and the structure of the application to use it. 5 of the 69 participant stated that they would need the assistance of an expert or a technical person, in contrast to 62 participants who would not require technical support. Item seven addresses the skills of other users rather than the own skills of the participant. 69% assume that other users would learn the system very quickly, while 13% of the participants think that other users would require a certain learning period. Several participants also stated that the experience with the desktop application helped them during the performance of the five tasks.

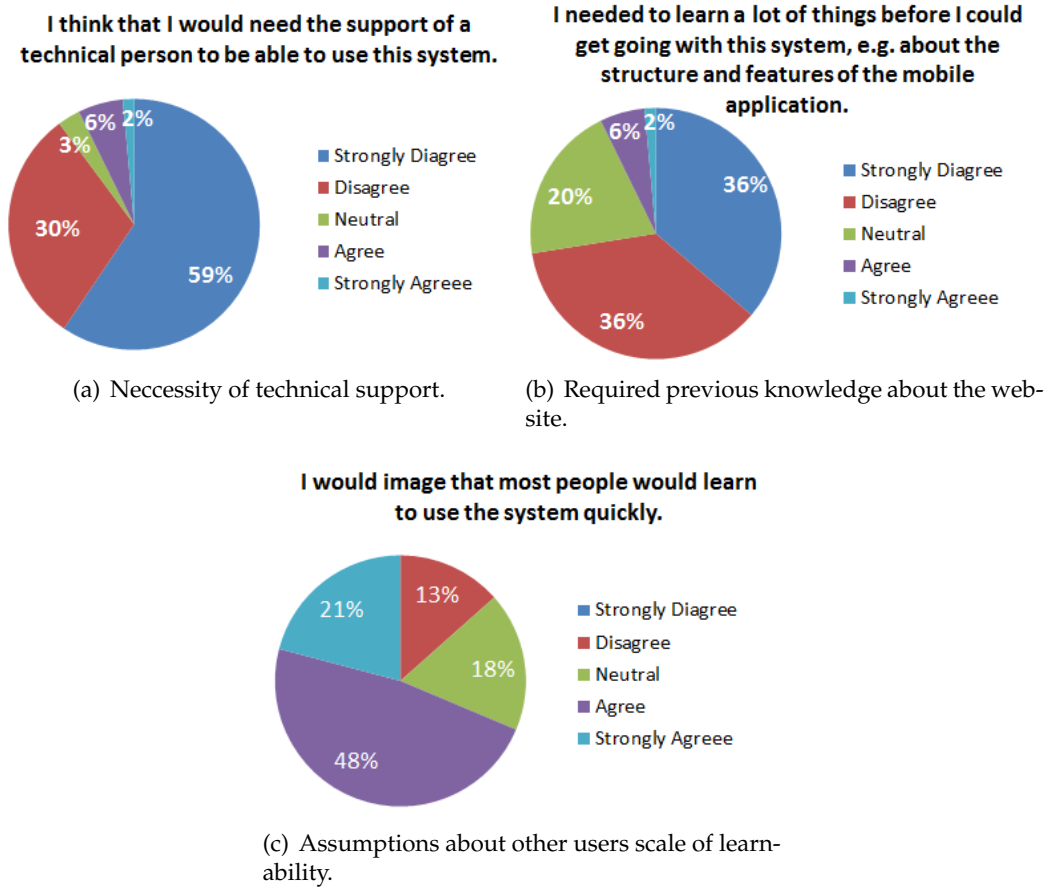


Figure 8.5.: Results regarding the learnability of the website.

The SUS score

The result of the SUS questionnaire is the SUS score, which is calculated with the responses of the questionnaire. Thereby, each response is assigned a scale from 1 to 5 where 1 represents "Strongly disagree" and 5 represents "Strongly agree". The score can range from 0 to 100, which represents the best achievable result. However, the score does not represent percentages but should be considered in terms of a percentile ranking. A SUS score of 68 is considered as average [55]. Several authors provided interpretations of the SUS score in terms of product usability. After analyzing over 2000 SUS surveys Bangor et al. [2] developed a scale to compare school grades, adjective ratings and accessibility scores with ranges of the SUS score. The rankings can be seen in Figure 8.6.

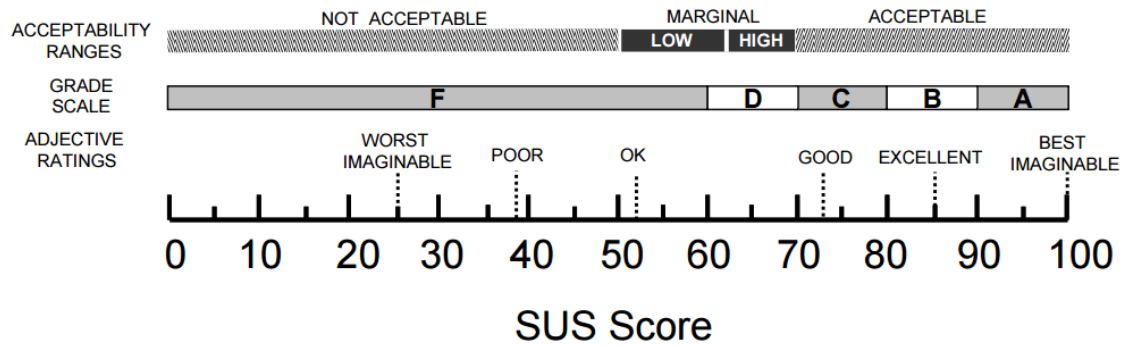


Figure 8.6.: A comparison of adjective ratings, grade scales and acceptability ranges in relation to the average SUS score.

Sauro [55] provides a percentile ranking of SUS scores, applying the grade ranking from Bangor et al. His ranking is illustrated in Figure 8.7.

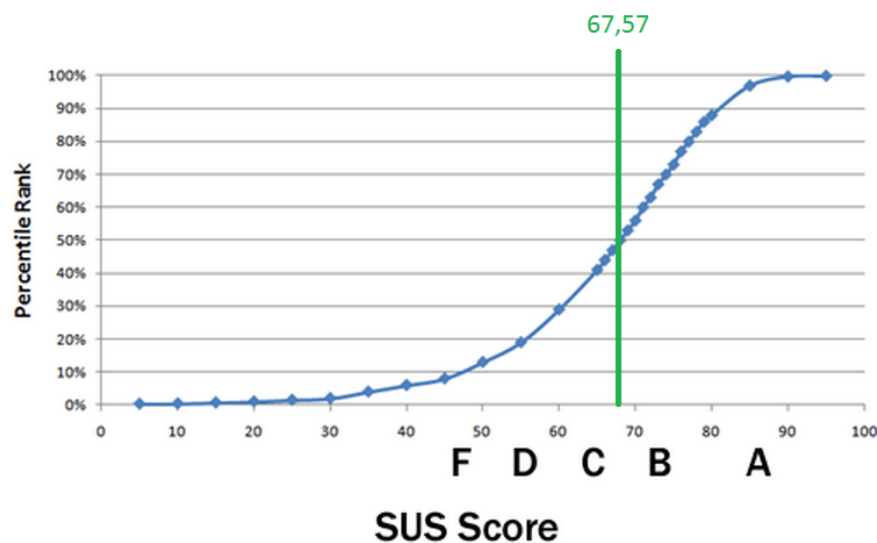


Figure 8.7.: The relation of the SUS score and school grades including the achieved SUS score from the user evaluation.

The developed mobile website received an average SUS score of 67,57 and falls at the 50th percentile according to Sauro. This indicates that the score is higher than 50% of all tested applications and results in a grade of C in terms of usability. According to Bangor et al. the acceptability of the system it is classified as highly marginal.

8.2.7. Discussion

The results from the empirical evaluation show that certain parts of the mobile website need to be refined in order to improve the overall system usability. The observations of the Think-Aloud protocol help to interpret the results of the questionnaire. The SUS score indicates that the usability of the system received an average rating from the users. All participants were able to successfully create and edit tasks with the mobile system. Certain users were also able to apply their experience and knowledge about the desktop application to the mobile use case. However, it should be considered that the use of a desktop browser to emulate the mobile interface can have influenced the perception of the participants. Because desktop browsers do not simulate real touch interactions, these inputs can be difficult to test. Hence, additional evaluations in the future with mobile devices can serve further information regarding user and device interaction.

Based on the findings of the evaluation, several improvements regarding the design and functionality were performed. On the website the user can see tasks on various pages, for example on the profile page, the alert page or the task overviews. To improve the accessibility to the tasks, a user can now edit tasks directly from these pages. Only on wiki pages the user still has to use the edit button. This aspect was not changed because users still may need to see assigned attributes of tasks. Furthermore, the selection and deselection of the filters was improved. An inactive filter is made active by clicking on it and vice versa, when the user clicks on an active filter he deactivates it.

On the website tabs are UI elements users will regularly encounter, e.g. on the profile page or the wiki pages. Therefore, several adjustments have been performed to the design to better highlight them as actionable elements and more effectively draw the user's focus and attention on them. The tabs have now a strong green background color against a grey background of the page. This helps to improve the visibility of the tabs by bringing them forward. The changes can be seen in Figure 8.8.

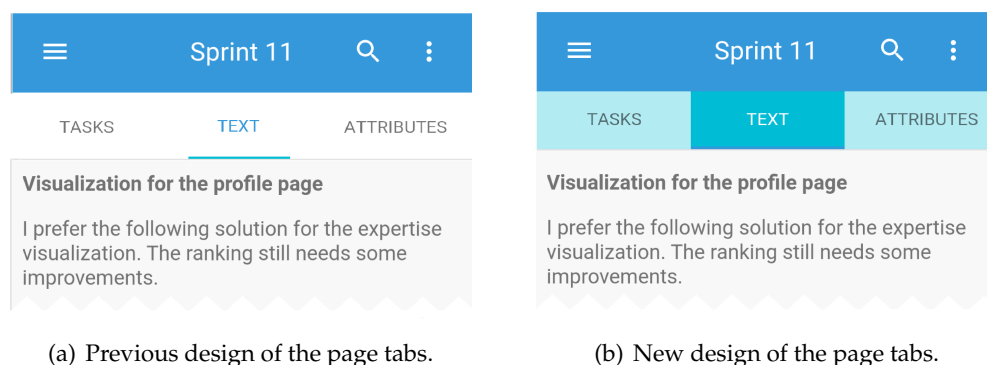


Figure 8.8.: Changes of the tab design on the wiki page.

Also the design of a number of buttons was adjusted to better distinct clickable content from not-clickable content. On the wiki page, the button which opens and closes the accordion view of completed tasks has been increased in size and has now a blue background color. A new drop-down icon indicates that by clicking the button additional content will be exposed. The new button design can be seen in Figure 8.9(b).

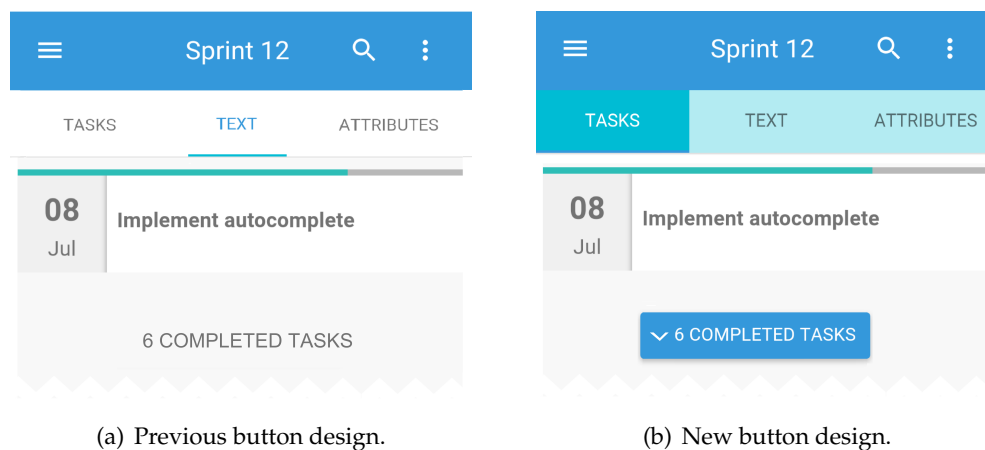


Figure 8.9.: Design of the 'completed tasks'-button.

Several aspects of the feed were also refined according to the feedback of the participants. The accordion view inside a feed entry comprising the like and comment button was removed. The user needed to click on the button with the three horizontal points to open and close the view (see Figure 8.10(a)). Now the like and comment button are immediately visible. Figure 8.10(b) shows the refined design without the accordion view.

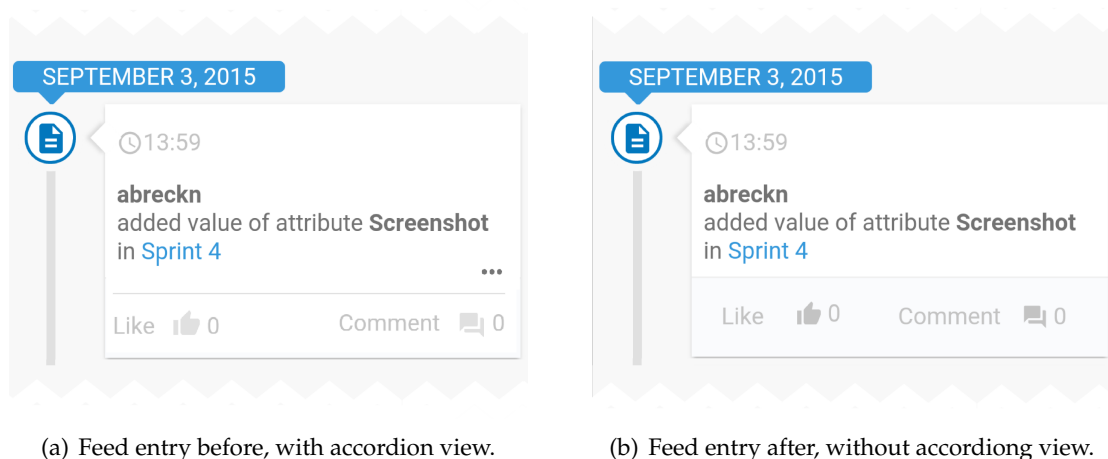


Figure 8.10.: Changes of the feed entry design.

Furthermore, the design of the filter buttons on the activity feed was changed to more conventional shapes. The buttons were adopted to the design of the desktop application to give them a more button-like shape. The similar design also helps the user to better recognize the buttons and increases the consistency between the two systems. The changes can be seen in Figure 8.11.

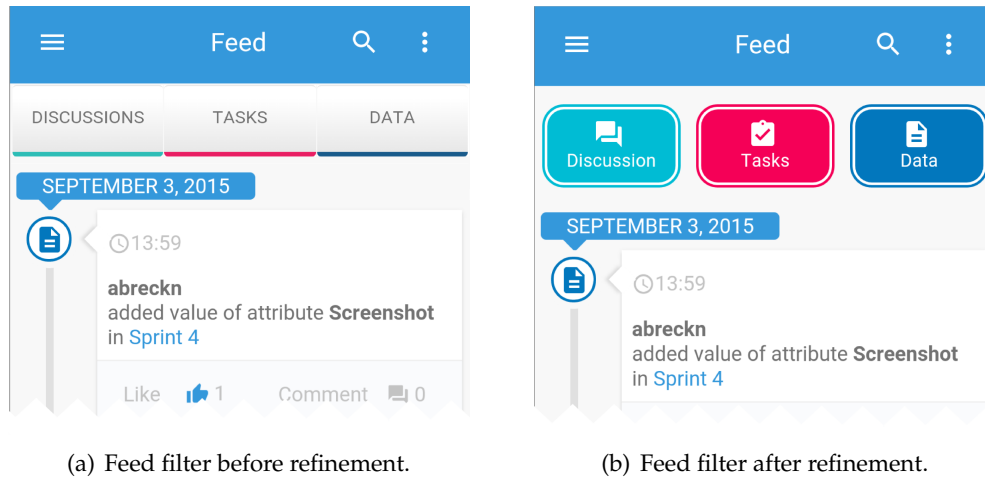


Figure 8.11.: Design changes of the feed filter.

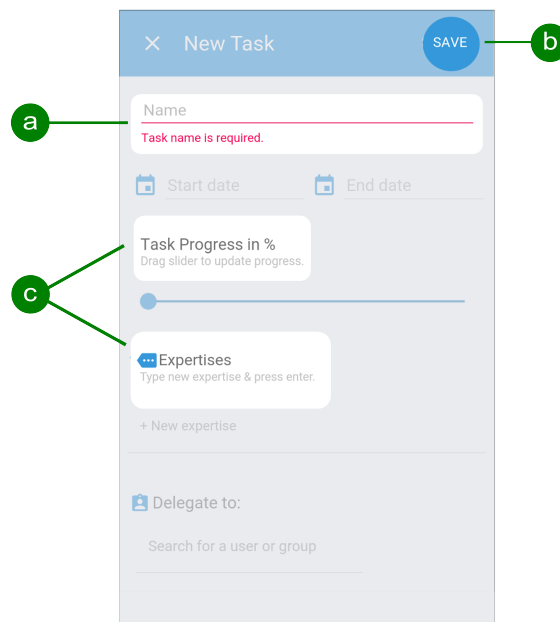


Figure 8.12.: Improvements to forms.

The application features several forms, e.g. to create and edit tasks or attributes. The initial forms were extended with informational content (see Figure 8.12). All forms are being checked with validations to ensure that the user provides the necessary information. Required fields are marked and missing fields are highlighted with a warning color (a). Furthermore, concise hints were placed below the input fields to inform the user about how to provide the input (c). For example, the user is told to drag the slider in order to change the task progress or to press enter in order to add a new expertise chip. The button to save dialog content was changed from a check icon to text (b). This helps user to better understand the underlying action of the button.

Part IV.

Conclusion

9. Conclusion and Outlook

This chapter provides an overview of the conducted work and presents the overall implications of this work's findings. Section 9.1 summarizes the key results of this thesis and discusses the novelty value and the research contributions. Section 9.2 gives a brief outlook to future work in this field of research.

9.1. Summary

The goal of this thesis was to develop a mobile solution which supports users to collaboratively structure knowledge-intensive processes. The current industrial environment is characterized by the rise of knowledge-centric organizations which aim to increase their innovation capabilities and strengthen their position on the global market. The combination of the areas knowledge management and mobile computing enhances knowledge distribution through collaborative networks and creates new prospects for organizations to benefit from advances in mobile technology. Darwin Mobile provides a work environment where knowledge workers can structure their processes within easily manageable tasks. They furthermore can create, manage and acquire knowledge, share their knowledge within a community and improve collaborative work based on experiences.

The development of the system relied on a user-centered design approach to focus on the end-user's perspective and needs. The involvement of users in the stages of the development cycle helped to determine the feature set of the system and identify improvements where necessary. The relevant use cases for the mobile interface were determined by identifying the characteristics of knowledge-intensive processes and the requirements to manage them. An extensive review of existing solutions and conducted research regarding mobile support for knowledge work helped to define the final feature set for the mobile web application.

The system was developed to comply with well-established design and quality standards from literature and research. With mobile devices having reached a mainstream status it is as ever important to understand the characteristics of mobile devices and how they differ from traditional desktop computers. Therefore, to create an engaging mobile solution for knowledge workers usability requirements towards mobile design based on usability standards and guidelines were studied and implemented.

An evaluation with students was conducted in the early stages of the development process to identify issues regarding the usability of the system and measure the overall user satisfaction. The evaluation showed that non-expert users were able to successfully complete

knowledge-intensive tasks with the mobile system without prior experience. The assessment of the system further revealed several possible improvements to the system design which were later on implemented in the ongoing design process.

9.2. Outlook

The research in mobile support for knowledge-intensive processes is still in early stages. Solutions providing adequate mobile collaboration and management structures for knowledge workers from academic research are still limited. However, with mobile knowledge work as a fast growing segment and continuous advances in mobile technology, new insights in this field of research will most likely emerge.

The developed mobile web solution focuses on the aspects tasks, content organization and collaboration between knowledge workers. It thereby aims to complement the existing desktop application Darwin, but does not yet implement all of its features. Therefore, the current system still leaves room for additional functionality.

With tasks and attributes as main building blocks of collaborative knowledge work, they can still be extended with several features to provide users with additional means to structure their processes. For now users can only create new attributes, but cannot define their values. Therefore, in the future users should be able to add values to different types of attributes. This includes uploading files or even directly using the mobile phone's camera to take pictures. Furthermore, users should be able to download attribute files as well to view them on the mobile device and always have them at hand. Another extension would be the possibility to assign attributes to tasks since in many cases attributes represent work results of tasks.

In the conducted user evaluation participants frequently stated that they wished a more personalized view of the system which would still allow them to collaborate in teams. Therefore, both aspects collaboration and personalization could be enhanced in several ways. To improve the collaboration between users, additional features of social networks could be realized. By tagging users for example in comments or posts team members could directly address other team members in the feed.

For now, the web application only supports the creation of public tasks. While reading and writing rights for wikis and attributes can be defined, this feature is not yet available for tasks in the current version. This means that all created tasks are visible to all users even if they are assigned to specific users or groups. In some cases, however, sensitive tasks and content should be protected or only visible for the task owners or task assignees. The user evaluation as well revealed the importance of privacy for system users. In order to create protected areas for users, newly created tasks could initially only be visible for the task owner. He can then configure additional privacy setting and permission.

To further provide a more personalized workspace wikis that are relevant for the user could be better highlighted. The navigation for example can become very extensive with a growing amount of wikis. Therefore, a user should only see the wikis he is favored or is active in. Other wikis should be displayed only on demand.

Due to the restricted period of time for this thesis the developed mobile web application was only tested with a selected number of devices and browsers. However, to provide a production ready system which is fully functional and consistent across all devices and platforms extensive testing is required. As an important part of quality assurance, cross-browser testing can help to uncover potential discrepancies between mobile devices and browsers.

Appendix

A. Darwin Mobile Web Application Overview

A1 Login Page

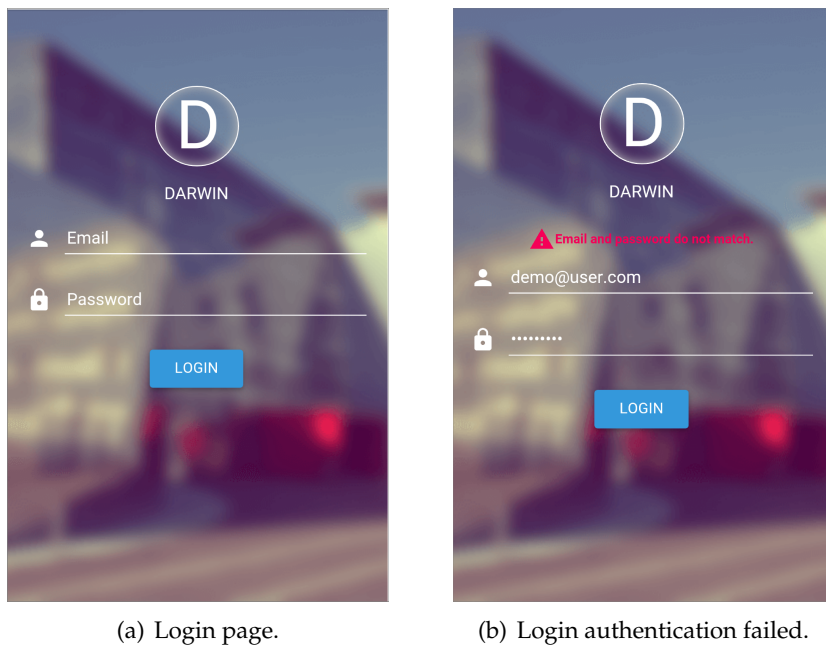
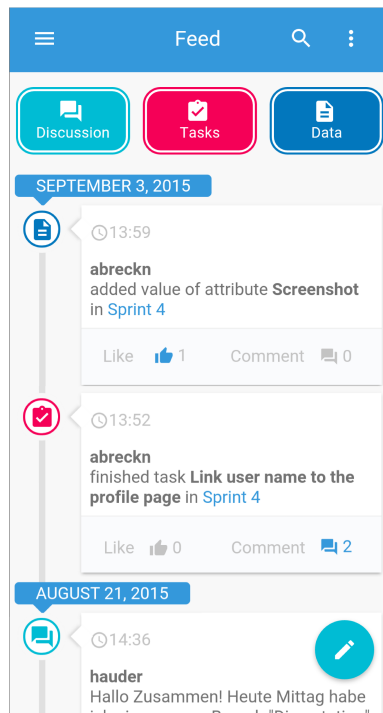
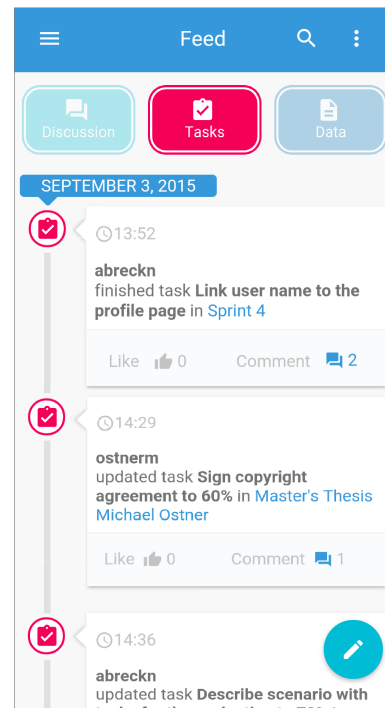


Figure A.1.: Overview of the login page.

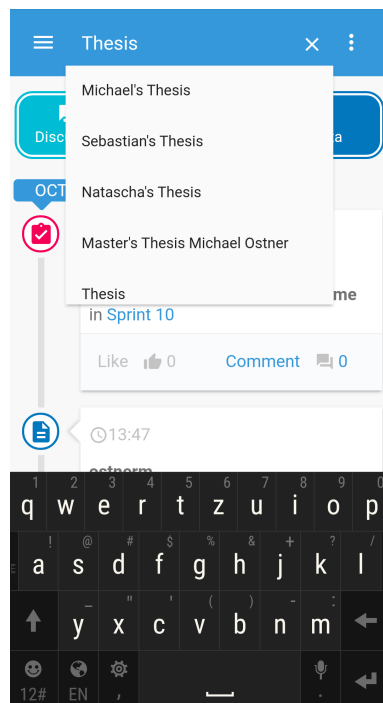
A2 Feed



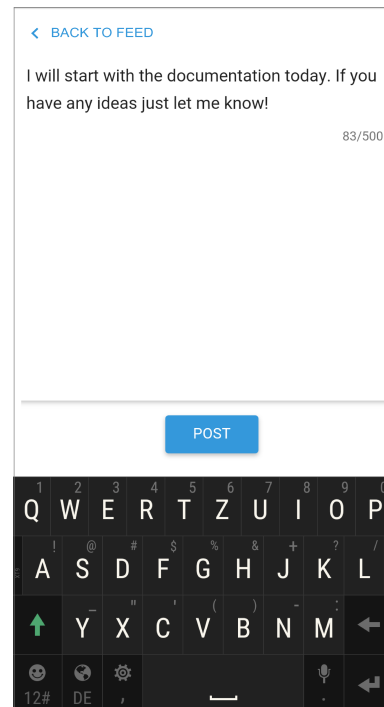
(a) Overview of different feed entry types.



(b) Filtered feed entries.



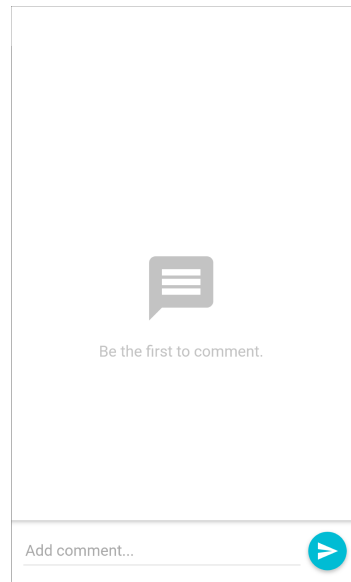
(c) Searching for wikis and wiki pages.



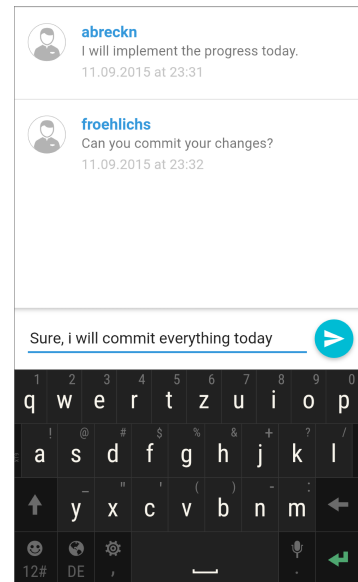
(d) Creating a new discussion post.

Figure A.2.: Overview of feed components.

A3 Feed Entry Comments



(a) First comment for a feed.



(b) Overview existing comments.

Figure A.3.: Overview of the comment feature.

A4 Sidebar Navigation

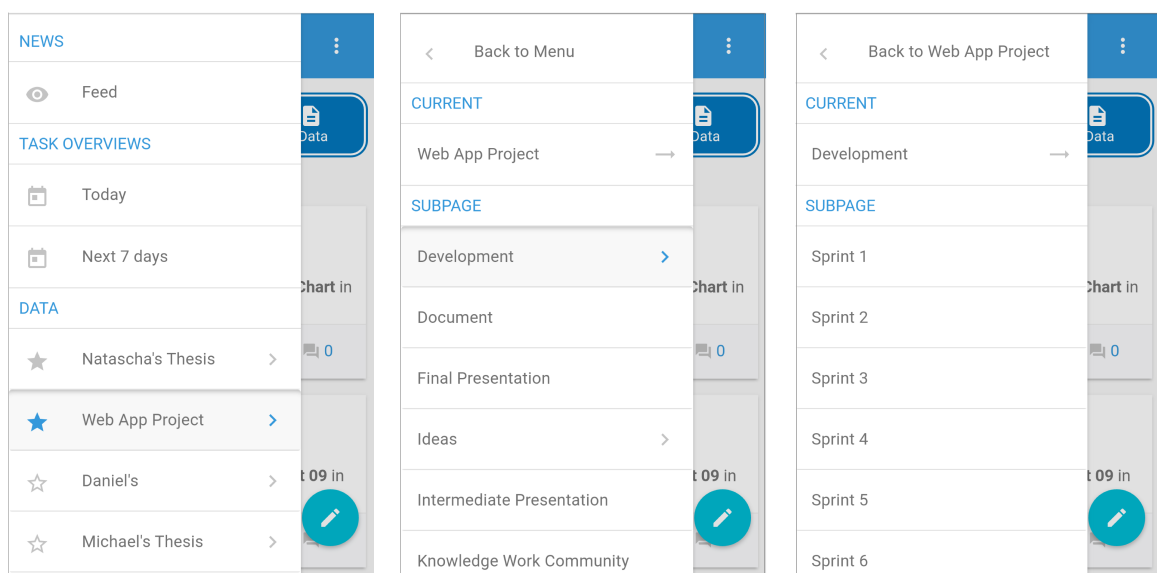


Figure A.4.: Sidebar drawer navigation with three levels.

A5 Drop-Down Menu

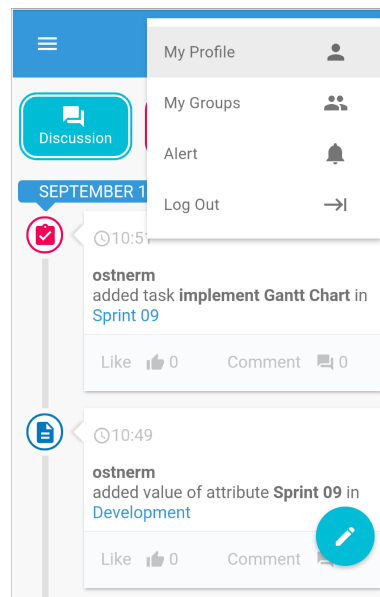
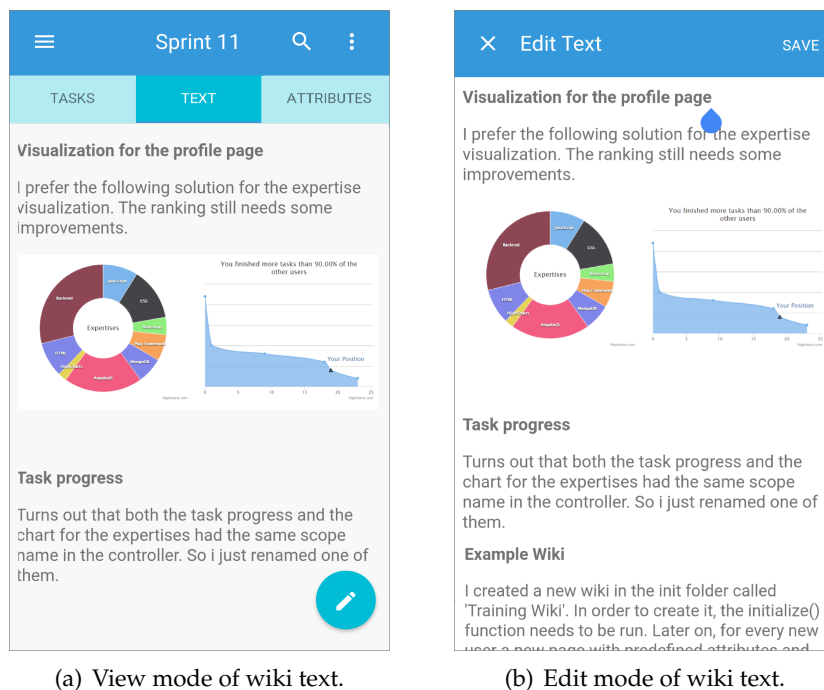


Figure A.5.: Drop-down menu.

A6 Wiki Page Text

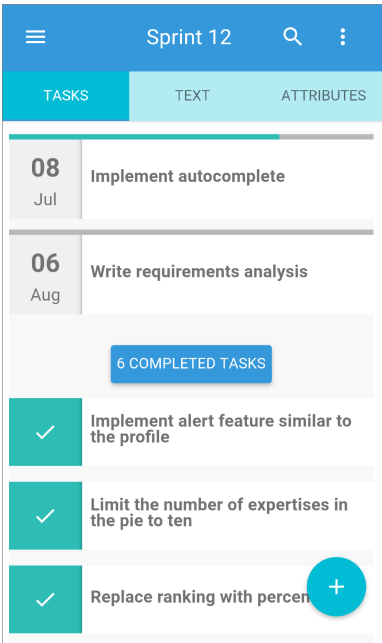


(a) View mode of wiki text.

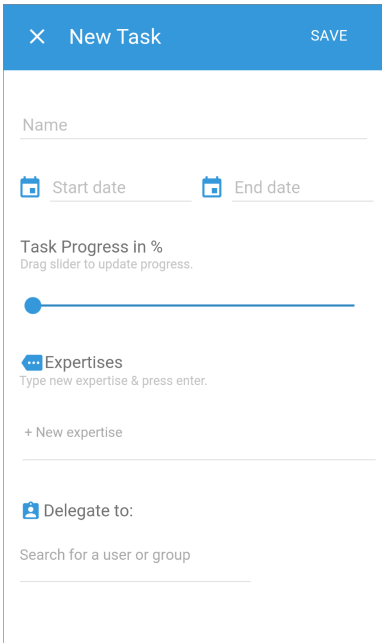
(b) Edit mode of wiki text.

Figure A.6.: Text of wiki page.

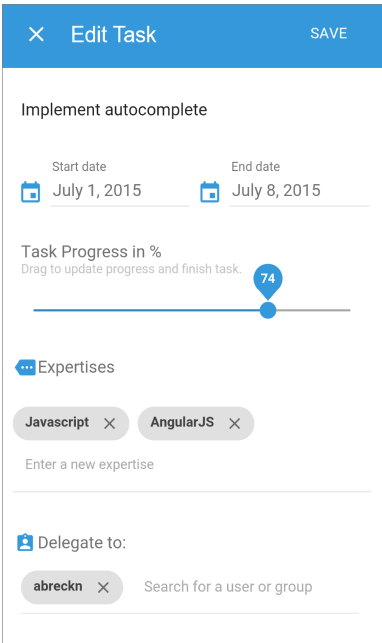
A7 Wiki Page Tasks



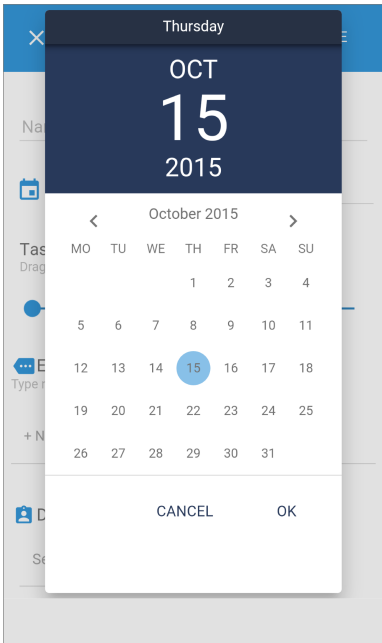
(a) Task overview.



(b) Creating a new task.



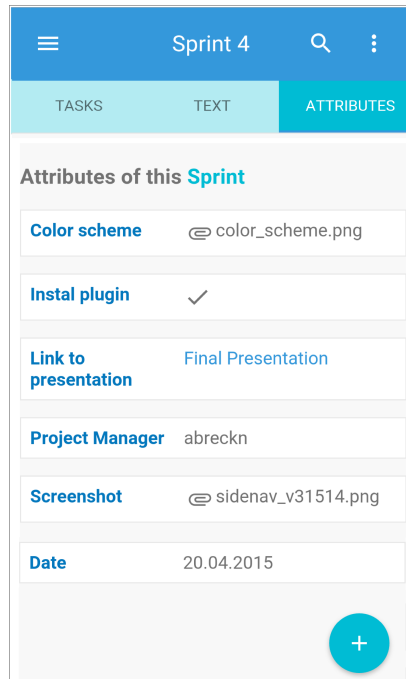
(c) Editing a task.



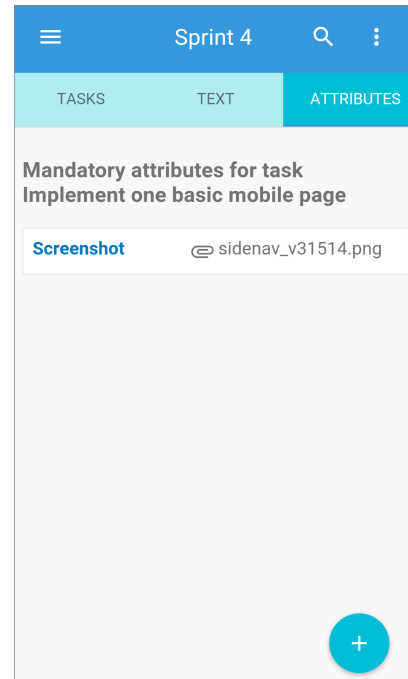
(d) Datepicker for setting start and end date.

Figure A.7.: Overview of the task features.

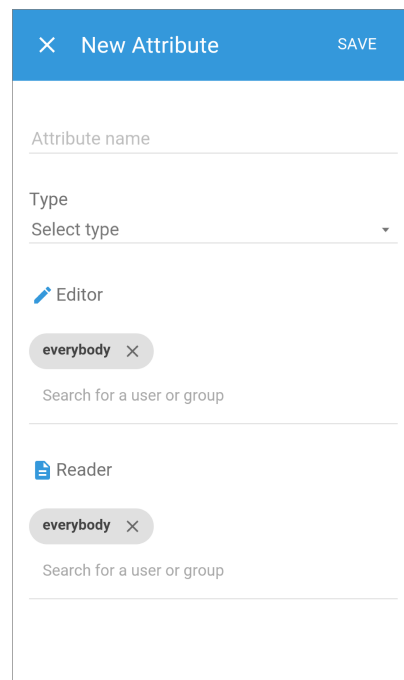
A8 Wiki Page Attributes



(a) Attributes list with different types.



(b) Mandatory attribute for selected task.



(c) Dialog for new attribute.

Figure A.8.: Components of the attribute overview.

A9 Creating new objects

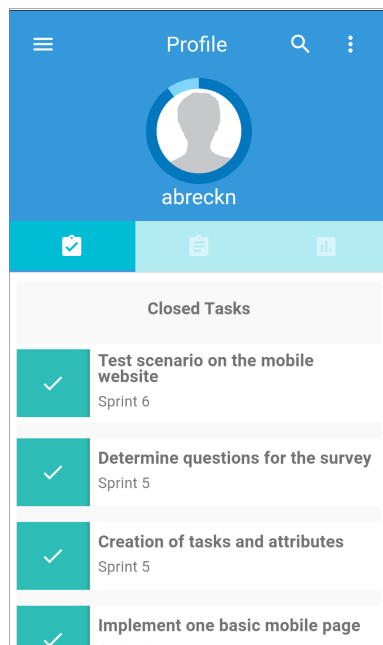
The figure shows two screenshots of a 'New Object' dialog box. Both have a blue header with a close button (X), the title 'New Object', and a 'SAVE' button.

(a) Creating a new wiki. The dialog box has a dropdown menu labeled 'Create a new...' with 'Wiki' selected. Below it is a text input field labeled 'Name'.

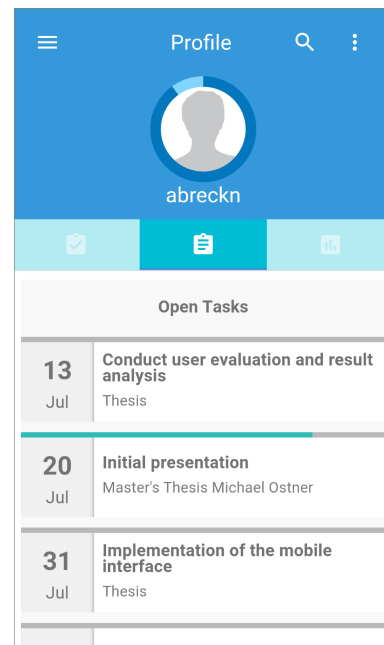
(b) Creating a new wiki page. The dialog box has a dropdown menu labeled 'Create a new...' with 'Page' selected. Below it are three text input fields: 'Name', 'Page type', and 'Subpage of'. The 'Subpage of' field has a placeholder text 'Search for existing wiki or page.' and the value 'Natascha's Thesis'.

Figure A.9.: Creating new objects.

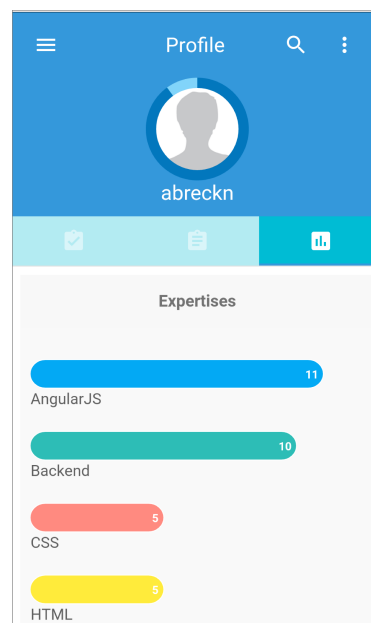
A10 Profile Page



(a) Closed tasks.



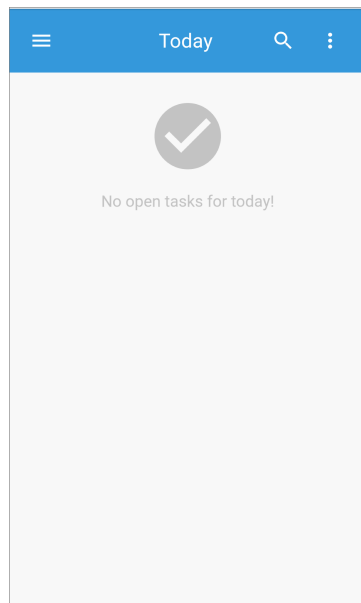
(b) Open tasks.



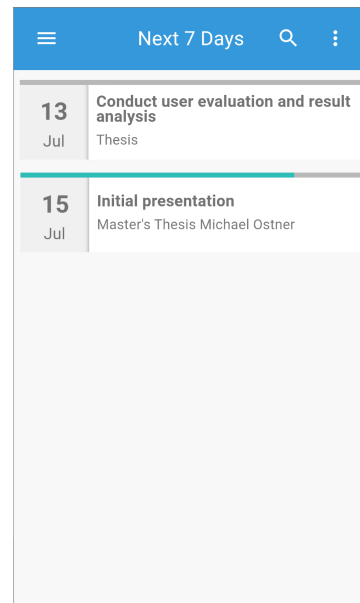
(c) User expertises.

Figure A.10.: Components of the profile page.

A11 Task Overview Pages



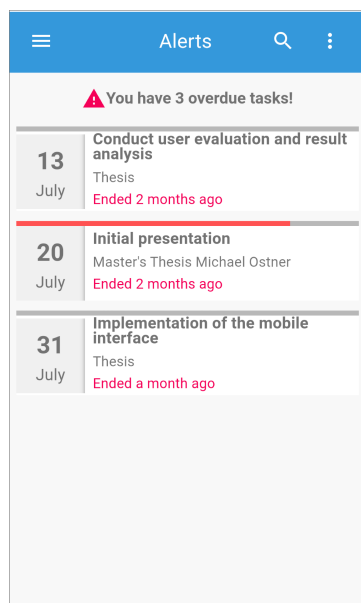
(a) Tasks of the current day.



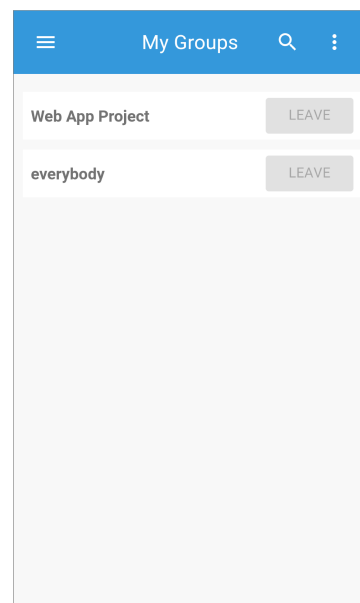
(b) Tasks of the next seven days.

Figure A.11.: Task overviews.

A12 Alert and Group Page



(a) Overview of due tasks.



(b) Overview of user's groups.

Figure A.12.: Alert and group page.

Bibliography

- [1] Sara Albolino, Mina Distratis, Thomas Schael, and Gloria Sciarra. Mobile knowledge worker. In *5th International Conference, Organizational Learning and Knowledge, Lancaster University Management School*. Citeseer, 2003.
- [2] Aaron Bangor, Philip Kortum, and James Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3):114–123, 2009.
- [3] Christian Becker and Christian Bizer. Exploring the geospatial semantic web with dbpedia mobile. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):278–286, 2009.
- [4] N Bevan. Kirakowski and j., maissel, j. 1991. what is usability. In *Proceeding of the Fourth International Conference on HCI*, 1991.
- [5] Ian Brinkley. Defining the knowledge economy. *London: The work foundation*, page 19, 2006.
- [6] John Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
- [7] John Brooke. Sus: a retrospective. *Journal of Usability Studies*, 8(2):29–40, 2013.
- [8] Giles Colborne. *Simple and usable web, mobile, and interaction design*. New Riders, 2010.
- [9] World Wide Web Consortium et al. Mobile web best practices 1.0. *Disponível online em <http://www.w3.org/TR/mobile-bp>*, 2008.
- [10] Joëlle Coutaz. Evaluation techniques: Exploring the intersection of hci and software engineering. In *Software Engineering and Human-Computer Interaction*, pages 35–48. Springer, 1995.
- [11] Thomas H Davenport and James E Short. Information technology and business process redesign. *Operations management: critical perspectives on business and management*, 1:97, 2003.
- [12] Claudio Di Ciccio, Andrea Marrella, and Alessandro Russo. Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches. *Journal on Data Semantics*, 4(1):29–57, 2014.
- [13] ISO DIS. 9241-210: 2010. ergonomics of human system interaction-part 210: Human-centred design for interactive systems. *International Standardization Organization (ISO). Switzerland*, 2009.

- [14] Peter F Drucker. Management: Tasks, responsibilities, practices. *Responsibilities, Practices*, 523, 1974.
- [15] Peter F Drucker. Knowledge-worker productivity: The biggest challenge. *The knowledge management yearbook 2000-2001*, 1999.
- [16] K Anders Ericsson and Herbert A Simon. Verbal reports as data. *Psychological review*, 87(3):215, 1980.
- [17] Timofey Ermilov, Norman Heino, Sebastian Tramp, and Sören Auer. Ontowiki mobile-knowledge management in your pocket. In *The Semantic Web: Research and Applications*, pages 185–199. Springer, 2011.
- [18] Adrian Fernandez, Emilio Insfran, and Silvia Abrahão. Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, 53(8):789–817, 2011.
- [19] Maximiliano Firtman. *Programming the mobile web*. " O'Reilly Media, Inc.", 2010.
- [20] Brian Fling. *Mobile Design and Development: Practical concepts and techniques for creating mobile sites and web apps*. " O'Reilly Media, Inc.", 2009.
- [21] Lyza Danger Gardner and Jason Grigsby. *Head First Mobile Web*. " O'Reilly Media, Inc.", 2011.
- [22] Sabrina Geiger. *Konzeption und Entwicklung einer auf Smartphones optimierten mobilen Anwendung für kollaboratives Checklisten-Management*. PhD thesis, Ulm University, 2013.
- [23] Matthias Gerber. *Konzeption und Realisierung fortschrittlicher aufgabenbasierter Koordinationsverfahren in mobilen Anwendungen*. PhD thesis, Ulm University, 2015.
- [24] Jun Gong and Peter Tarasewich. Guidelines for handheld mobile device interface design. In *Proceedings of DSI 2004 Annual Meeting*, pages 3751–3756. Citeseer, 2004.
- [25] Google. Material design.
- [26] Kim Griggs, Laurie M Bridges, and Hannah Gascho Rempel. Library/mobile: tips on designing and developing mobile web sites. *Code4lib journal*, 8, 2009.
- [27] Matthias Grimm, Mohammad-Reza Tazari, and Dirk Balfanz. A reference model for mobile knowledge management. In *Proceedings of I-KNOW*, volume 5, pages 54–62, 2005.
- [28] Mordechai Haklay. *Interacting with geospatial technologies*. Wiley Online Library, 2010.
- [29] Monty Hammontree, Paul Weiler, and Nandini Nayak. Remote usability testing. *Interactions*, 1(3):21–25, 1994.
- [30] Matheus Hauder, Rick Kazman, and Florian Matthes. Empowering end-users to collaboratively structure processes for knowledge work. In *Business Information Systems*, pages 207–219. Springer, 2015.

- [31] Henning Heitkötter, Tim A Majchrzak, Benjamin Ruland, and Till Weber. Evaluating frameworks for creating mobile web apps. In *WEBIST*, pages 209–221, 2013.
- [32] Peter Hilton, Erik Bakker, and Francisco Canedo. *Play for Scala: Covers Play 2*. Manning Publications Co., 2013.
- [33] ISO. Standards.
- [34] ISO. 9241-11:1998, ergonomic requirements for office work with visual display terminals (vdt), part 11: Guidance on usability. *International Standardization Organization (ISO)*. Switzerland, 1998.
- [35] ISO/IEC. Iso/iec. 13407, human-centred design processes for interactive systems. *International Standardization Organization (ISO)*. Switzerland, 1999.
- [36] ISO/IEC. 9126, software product evaluation - quality characteristics and guidelines for the user. *International Standardization Organization (ISO)*. Geneva, 2001.
- [37] Julie A Jacko. *Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*. CRC press, 2012.
- [38] Axel Korthaus and Martin Schader. Using a topic grid and semantic wikis for ontology-based distributed knowledge management in enterprise software development processes. In *Enterprise Distributed Object Computing Conference Workshops, 2006. EDOCW'06. 10th IEEE International*, pages 4–4. IEEE, 2006.
- [39] Uday Kulkarni and Minu Ipe. Decision support for knowledge intensive business processes. In *Proceedings of the ADPSI Conference*, 2007.
- [40] Marcello La Rosa, Marlon Dumas, Arthur HM Ter Hofstede, and Jan Mendling. Configurable multi-perspective business process models. *Information Systems*, 36(2):313–340, 2011.
- [41] Fritz Machlup. *The production and distribution of knowledge in the United States*, volume 278. Princeton university press, 1962.
- [42] Cameron Moll. *Mobile web design*. Lulu. com, 2008.
- [43] Nicolas Mundbrod, Jens Kolb, and Manfred Reichert. Towards a system support of collaborative knowledge work. In *Business process management workshops*, pages 31–42. Springer, 2013.
- [44] Nicolas Mundbrod and Matthew Reichert. Process-aware task management support for knowledge-intensive business processes: Findings, challenges, requirements. In *Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), 2014 IEEE 18th International*, pages 116–125. IEEE, 2014.
- [45] Jakob Nielsen. *Usability engineering*. Elsevier, 1994.
- [46] Jakob Nielsen. Usability inspection methods. In *Conference companion on Human factors in computing systems*, pages 413–414. ACM, 1994.

- [47] Jakob Nielsen and Raluca Budiu. *Mobile usability*. MITP-Verlags GmbH & Co. KG, 2013.
- [48] Klaus North and Stefan Gueldenberg. *Effective knowledge work: Answers to the management challenges of the 21st century*. Emerald Group Publishing, 2011.
- [49] Martin Pfiffner and Peter Stadelmann. *Wissen wirksam machen: wie Kopfarbeiter produktiv werden*. Campus Verlag, 2012.
- [50] Jenny Preece, Helen Sharp, and Yvonne Rogers. *Interaction Design-beyond human-computer interaction*. John Wiley & Sons, 2015.
- [51] Manfred Reichert and Barbara Weber. *Enabling flexibility in process-aware information systems: challenges, methods, technologies*. Springer Science & Business Media, 2012.
- [52] Mary Beth Rosson and John Millar Carroll. *Usability engineering: scenario-based development of human-computer interaction*. Morgan Kaufmann, 2002.
- [53] Roland T Rust, Debora Viana Thompson, and Rebecca W Hamilton. Defeating feature fatigue. *Harvard business review*, 84(2):98, 2006.
- [54] Timo Saari, Kari Kallinen, Mikko Salminen, Niklas Ravaja, and Marco Rapino. A mobile and desktop application for enhancing group awareness in knowledge work teams. In *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction*, pages 95–104. Springer, 2009.
- [55] Jeff Sauro. Measuring usability with the system usability scale (sus). 2011.
- [56] Sebastian Schaffert. Ikewiki: A semantic wiki for collaborative knowledge management. In *Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2006. WET-ICE'06. 15th IEEE International Workshops on, pages 388–396. IEEE, 2006.
- [57] Helen Schonenberg, Ronny Mans, Nick Russell, Nataliya Mulyar, and Wil van der Aalst. Process flexibility: A survey of contemporary approaches. In *Advances in Enterprise Engineering I*, pages 16–30. Springer, 2008.
- [58] Ben Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*, volume 3. Addison-Wesley Reading, MA, 1992.
- [59] Elizabeth A Smith. The role of tacit and explicit knowledge in the workplace. *Journal of knowledge Management*, 5(4):311–321, 2001.
- [60] Daniel Sonntag, Ralf Engel, Gerd Herzog, Alexander Pfalzgraf, Norbert Pfleger, Massimo Romanelli, and Norbert Reithinger. Smartweb handheld-multimodal interaction with ontological knowledge bases and semantic web services. In *Artificial Intelligence for Human Computing*, pages 272–295. Springer, 2007.
- [61] Nico Stehr. Knowledge societies. 1994.
- [62] Stefan Stieglitz, Christoph Lattemann, and Tobias Brockmann. Mobile applications for knowledge workers and field workers. *Mobile Information Systems*, 2015, 2015.

- [63] Moh-Reza Tazari, Lukas Windlinger, and Thomas Hoffmann. Knowledge management requirements of mobile work on information technology. *Mobile Work Employs IT (MoWeIT'05), Prague*, 2005.
- [64] Toilal. angular-material-components.
- [65] Tom Tullis, Stan Fleischman, Michelle McNulty, Carrie Cianchette, and Margaret Bergel. An empirical comparison of lab and remote usability testing of web sites. In *Usability Professionals Association Conference*, 2002.
- [66] Roman Vaculin, Richard Hull, Terry Heath, Craig Cochran, Anil Nigam, and Piyawadee Sukaviriya. Declarative business artifact centric modeling of decision and knowledge intensive business processes. In *Enterprise Distributed Object Computing Conference (EDOC), 2011 15th IEEE International*, pages 151–160. IEEE, 2011.
- [67] Stefan Voigt, Frank Fuchs-Kittowski, and Andreas Gohr. Structured wikis: Application oriented use cases. In *Proceedings of The International Symposium on Open Collaboration*, page 22. ACM, 2014.
- [68] Christian Wagner. Wiki: A technology for conversational knowledge management and group collaboration. *The Communications of the Association for Information Systems*, 13(1):58, 2004.
- [69] James P Ware and Charles E Grantham. Knowledge work and knowledge workers. *WIRED West Michigan*, 2007.
- [70] Scott Weiss. *Handheld usability*. John Wiley & Sons, 2003.
- [71] Max L Wilson, Alistair Russell, Daniel A Smith, Alisdair Owens, et al. mspace mobile: A mobile application for the semantic web. 2005.
- [72] Yufei Yuan and Wuping Zheng. Mobile task characteristics and the needs for mobile work support: a comparison between mobile knowledge workers and field workers. In *Mobile Business, 2009. ICMB 2009. Eighth International Conference on*, pages 7–11. IEEE, 2009.
- [73] Hong Zhan, Tian Tang, Yue Zhang, et al. The research on characteristics of knowledge workers and their motivating factors: A review. *American Journal of Industrial and Business Management*, 3(06):557, 2013.

List of Figures

1.1. Research questions.	5
1.2. Outline of the thesis.	6
2.1. The collaborative KIP life cycle [43].	11
2.2. Components of a KIP [12].	13
2.3. The diagnostic-therapeutic cycle of patient treatment [44]	15
3.1. Definition of usability and quality of use.	18
3.2. Dimensions of usability and user experience.	19
3.3. Product- and process-oriented standards.	23
4.1. The mobile knowledge management concept by Grimm et al. [27].	27
4.2. OntoWiki Mobile. a) Overview of meta data. b) Editing of meta data. c) Adding images [17].	28
4.3. proCollab.	29
4.4. Any.Do.	30
4.5. Wunderlist.	31
4.6. Trello.	32
4.7. Todoist.	33
5.1. The user-centered design process according to ISO 13407 [35].	38
5.2. User-centered design process for Darwin Mobile.	39
5.3. User groups of Darwin and their characteristics [30].	40
5.4. Use cases for organizing content with wiki pages.	42
5.5. Use cases for managing attributes.	43
5.6. Use cases for managing tasks.	44
5.7. Use cases for the feed.	45
5.8. Use cases for enhanced visibility.	46
5.9. Use cases for the profile page.	46
6.1. System architecture.	55
6.2. Client and server architecture.	56
6.3. Example of a user agent header.	58
6.4. Template controller for dynamic serving.	58
6.5. Simplified class diagram of the datamodel.	59
6.6. Overview of the Darwin API and controller functions.	60
7.1. Structure of the mobile website.	61
7.2. Color schema of the mobile website.	62
7.3. Custom theming with Angular Material.	63

7.4. Login page.	63
7.5. Sidebar navigation drawer with three levels.	64
7.6. Drop-down menu.	65
7.7. Overview of feed components.	66
7.8. Concept of infinite scroll.	67
7.9. Infinite scroll for the feed.	68
7.10. Text of wiki page.	69
7.11. HTML code and Angular directive for editing wiki text.	69
7.12. Components of the task overview.	70
7.13. Components of the attribute overview.	71
7.14. Creating new objects.	72
7.15. Components of the profile page.	73
7.16. Overview of secondary pages.	74
8.1. The SUS questionnaire [6].	78
8.2. Results regarding the frequency of use.	82
8.3. Results regarding the ease of use of the website.	83
8.4. Results regarding the consistency of the website.	84
8.5. Results regarding the learnability of the website.	85
8.6. A comparison of adjective ratings, grade scales and acceptability ranges in relation to the average SUS score.	86
8.7. The relation of the SUS score and school grades including the achieved SUS score from the user evaluation.	86
8.8. Changes of the tab design on the wiki page.	87
8.9. Design of the 'completed tasks'-button.	88
8.10. Changes of the feed entry design.	88
8.11. Design changes of the feed filter.	89
8.12. Improvements to forms.	89
A.1. Overview of the login page.	99
A.2. Overview of feed components.	100
A.3. Overview of the comment feature.	101
A.4. Sidebar drawer navigation with three levels.	101
A.5. Drop-down menu.	102
A.6. Text of wiki page.	102
A.7. Overview of the task features.	103
A.8. Components of the attribute overview.	104
A.9. Creating new objects.	105
A.10. Components of the profile page.	106
A.11. Task overviews.	107
A.12. Alert and group page.	107

List of Tables

2.1. Differences between knowledge and manual work [48].	9
5.1. Differences between mobile knowledge workers and mobile field workers [62].	41
5.2. Overview of the framework assessment according to defined criteria.	50
8.1. Task description of the user evaluation.	79