



Universität Passau  
Fakultät für Informatik und Mathematik  
Lehrstuhl für Informatik mit Schwerpunkt Verteilte  
Informationssysteme  
Betreuer: Prof. Harald Kosch

# Prozessdesign- und Digitalisierung für eine moderne Wasserwachtverwaltung

Bachelorarbeit

18.01.2024

Name:	Felix, Wilhelm
Matrikelnummer:	78186
Fachsemester:	14
Studiengang:	Internet Computing
E-Mail Adresse:	wilhel40@ads.uni-passau.de

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>2</b>
<b>Abbildungsverzeichnis</b>	<b>4</b>
<b>1. Einleitung</b>	<b>5</b>
1.1. Hintergrund und Relevanz des Themas . . . . .	5
1.2. Zielsetzung der Arbeit . . . . .	5
1.3. Methodik und Vorgehensweise . . . . .	5
<b>2. Theoretischer Hintergrund</b>	<b>6</b>
2.1. Wasserwacht als gemeinnützige Organisation . . . . .	6
2.2. Herausforderungen bei der Wachplanerstellung . . . . .	7
2.3. Bedeutung von IT-Lösungen zur Digitalisierung von Verwaltungsprozessen . . . . .	8
2.4. Best Practices Websitedesign . . . . .	9
2.4.1. Usability . . . . .	9
2.4.2. Responsive Design . . . . .	9
<b>3. Anforderungsanalyse</b>	<b>11</b>
3.1. Identifikation von Anforderungen an die Webanwendung . . . . .	11
3.1.1. Nutzergruppen . . . . .	11
3.1.2. Wachplanung . . . . .	11
3.1.3. Wachbuch . . . . .	11
3.2. Befragung von Wasserwachtmitgliedern und Mockups . . . . .	12
3.2.1. Mockups . . . . .	12
3.2.2. Ergebnisse . . . . .	13
<b>4. Technologien und Methoden</b>	<b>20</b>
4.1. Auswahl geeigneter Technologien und Tools für die Webanwendung . . . . .	20
4.2. Beschreibung der verwendeten Programmiersprachen, Frameworks und Datenbanken . . . . .	20
4.2.1. Java . . . . .	20
4.2.2. Spring . . . . .	21
4.2.3. Thymeleaf . . . . .	21
4.2.4. Bootstrap . . . . .	22
4.2.5. Postgres & Spring Data JPA . . . . .	22
4.2.6. Wetter API . . . . .	23
4.3. Agile Entwicklungsmethoden zur Umsetzung des Projekts . . . . .	24

<b>5. Konzeption und Umsetzung</b>	<b>25</b>
5.1. Detaillierte Beschreibung der Konzeption der Webanwendung . . . . .	25
5.1.1. Datenmodell . . . . .	25
5.2. Funktionalitäten und Features der Anwendung . . . . .	28
5.2.1. Spring Security . . . . .	28
5.2.2. Anmelde- / Registrierungsprozess . . . . .	29
5.2.3. Navigationsleiste . . . . .	30
5.2.4. Nutzerübersicht . . . . .	31
5.2.5. Wachplanerstellung . . . . .	33
5.2.6. Wachplanübersicht . . . . .	34
5.2.7. Wachplandurchführung . . . . .	34
5.2.8. Druckfunktionalität . . . . .	38
<b>6. Evaluation</b>	<b>40</b>
6.1. Nutzertests . . . . .	40
6.1.1. Fragenkatalog . . . . .	40
6.1.2. Ergebnisse . . . . .	41
6.1.3. Diskussion . . . . .	42
6.2. Klickmetriken . . . . .	43
6.2.1. Wachplanerstellung . . . . .	43
6.2.2. Wachplandurchführung . . . . .	43
6.2.3. Interpretation . . . . .	44
6.3. Verbesserungspotentiale . . . . .	45
6.4. Ausblick auf zukünftige Entwicklungen und mögliche Erweiterungen der Webanwendung . . . . .	45
<b>7. Fazit</b>	<b>47</b>
7.1. Zusammenfassung der Arbeit . . . . .	47
7.2. Schlussfolgerungen und Handlungsempfehlungen . . . . .	47
<b>A. Anhang</b>	<b>49</b>
A.1. Initialer Entwurf . . . . .	49
A.2. Wachbuch Papier . . . . .	52
<b>Literaturverzeichnis</b>	<b>55</b>

# Abbildungsverzeichnis

3.1. AnmeldeProzess . . . . .	14
3.2. Freischaltungs-Prozess . . . . .	15
3.3. Wachplanung . . . . .	17
3.4. Wachbuch . . . . .	19
5.1. UML-Diagramm . . . . .	25
5.2. Auszug aus SecurityConfiguration.java . . . . .	28
5.3. Auszug aus UserController.java . . . . .	29
5.4. AnmeldeProzess . . . . .	30
5.5. Navigationsleisten . . . . .	31
5.6. Auszug aus header.html . . . . .	31
5.7. Nutzerübersicht . . . . .	32
5.8. Einzeltermin anlegen . . . . .	33
5.9. Terminreihe anlegen . . . . .	33
5.10. Wachtagsübersicht . . . . .	34
5.11. Wachtags Durchführung . . . . .	35
5.12. Wachtags mit gebuchten Helfern . . . . .	36
5.13. Wachtags mit anwesenden Helfern . . . . .	37
5.14. Auszug aus GuardDayService.java - Automatische Speicherung der Wetterdaten . . . . .	37
5.15. Wachende . . . . .	38
5.16. Wachbuch . . . . .	38
A.1. Initialer Entwurf . . . . .	51
A.2. Wachbuch Papier . . . . .	53
A.3. Wachtags Export . . . . .	54

# **1. Einleitung**

## **1.1. Hintergrund und Relevanz des Themas**

Die fortschreitende Digitalisierung und Automatisierung von Verwaltungsprozessen hat in den letzten Jahren auch vor gemeinnützigen Organisationen nicht haltgemacht. Insbesondere im Bereich der Wasserwacht, die sich der Rettung von Menschenleben in und um Gewässern widmet, eröffnen moderne Informationstechnologien neue Möglichkeiten, die Effizienz von Verwaltungstätigkeiten zu steigern. Im Rahmen meiner Bachelorarbeit habe ich mich mit diesem Thema auseinandergesetzt und eine Webanwendung entwickelt, um die Verwaltungsprozesse der Wasserwacht Dingolfing-Landau zu digitalisieren.

## **1.2. Zielsetzung der Arbeit**

Das Ziel dieser Arbeit ist es, die Herausforderungen bei der Wachplanerstellung zu identifizieren und durch den Einsatz von IT-Lösungen zu bewältigen. Dabei wird insbesondere der Prozess der Wachplanung betrachtet, der eine essenzielle Rolle in der Organisation der Wasserwacht einnimmt. Traditionell werden Wachpläne manuell erstellt, was mit einem erheblichen zeitlichem und administrativem Aufwand verbunden ist. Durch den Einsatz einer Webanwendung sollen diese Prozesse vereinfacht und effizienter gestaltet werden, um Zeit und Ressourcen einzusparen.

Die vorliegende Arbeit soll somit zur Verbesserung der Verwaltungsprozesse in der Wasserwacht beitragen und liefert einen Beitrag zur Nutzung moderner Informationstechnologien im gemeinnützigen Bereich. Es wird erwartet, dass die entwickelte Webanwendung die Planung und Organisation von Wacheinsätzen erleichtert und somit zur Optimierung der Einsatzbereitschaft der Wasserwacht beiträgt.

## **1.3. Methodik und Vorgehensweise**

Im Rahmen dieser Arbeit werden durch Interviews von Wasserwachtmitgliedern die Anforderungen an die Webanwendung analysiert, die Technologien zur Umsetzung ausgewählt und die Funktionalitäten der Anwendung im Detail erläutert. Die Evaluation der entwickelten Webanwendung erfolgt anhand von Nutzertests und Analyse von Klickmetriken, um die Praxistauglichkeit und Effektivität der Anwendung zu überprüfen.

## 2. Theoretischer Hintergrund

### 2.1. Wasserwacht als gemeinnützige Organisation

Die Wasserwacht ist eine gemeinnützige Organisation, die sich dem Schutz und der Sicherheit von Menschen im Wasser widmet. Als Teil des Deutschen Roten Kreuzes (DRK) setzt sie sich aus ehrenamtlichen Helfern und Fachkräften zusammen und spielt eine entscheidende Rolle bei der Gewährleistung der Wasserrettung und des Katastrophenschutzes.

Die Hauptaufgabe der Wasserwacht besteht darin, das Wohl und die Sicherheit von Menschen in Gewässern zu gewährleisten. Sie ist insbesondere in Seen, Flüssen, Schwimmbädern und Küstengebieten präsent und übernimmt wichtige Funktionen wie die Überwachung von Badebereichen, die Durchführung von Rettungsmaßnahmen und die Erste Hilfe bei Wasserunfällen. Die Wasserwacht ist darauf spezialisiert, in Notfällen schnell und effektiv zu reagieren und Menschenleben zu retten (vgl. [drkb]).

Neben der unmittelbaren Wasserrettung engagiert sich die Wasserwacht auch in der Prävention und Aufklärung. Sie organisiert Schwimmkurse, informiert über Gefahren im Wasser und beteiligt sich an Aktionen zur Förderung der Schwimmfähigkeit und Sicherheit im Wasser (vgl. [drka]). Durch diese Aktivitäten trägt die Wasserwacht dazu bei, Unfälle zu vermeiden und das Bewusstsein für den verantwortungsvollen Umgang mit Wasser zu schärfen.

Die Wasserwacht ist eng mit lokalen Gemeinschaften verbunden und arbeitet oft eng mit anderen Rettungsorganisationen, wie beispielsweise der Feuerwehr oder der DLRG, zusammen. Sie spielt eine wichtige Rolle bei der Bewältigung von Notfällen und Katastrophen, sei es bei Naturkatastrophen oder bei Großveranstaltungen in der Nähe von Gewässern (vgl. [brk]).

Als gemeinnützige Organisation ist die Wasserwacht auf die Unterstützung von Freiwilligen und Spenden angewiesen. Die ehrenamtlichen Helfer der Wasserwacht investieren ihre Zeit und ihr Engagement, um anderen Menschen in Notsituationen zu helfen und die Sicherheit in Gewässern zu gewährleisten. Sie absolvieren eine fundierte Ausbildung in Erster Hilfe, Rettungstechniken und Kommunikation, um bestmöglich auf Einsätze vorbereitet zu sein.

Insgesamt ist die Wasserwacht eine essenzielle gemeinnützige Organisation, die sich unermüdlich für die Sicherheit und das Wohlergehen von Menschen im Wasser einsetzt. Ihr Engagement, ihre Fachkenntnisse und ihre Einsatzbereitschaft machen sie zu einer vertrauenswürdigen Institution, die maßgeblich dazu beiträgt, Unfälle zu verhindern und Leben zu retten.

## 2.2. Herausforderungen bei der Wachplanerstellung

Die Wachplanerstellung in der Wasserwacht stellt eine Reihe von Herausforderungen dar, die es zu bewältigen gilt. Ein effizienter und gut durchdachter Wachplan ist von entscheidender Bedeutung, um sicherzustellen, dass zu jeder Zeit ausreichend qualifizierte Kräfte für die Gewässerüberwachung und Rettungseinsätze zur Verfügung stehen. In Gesprächen mit Wasserwachtmitgliedern wurden die Herausforderungen bei der Wachplanerstellung aufgezeigt:

**Verfügbarkeit der Helfer:** Die Wasserwacht ist in erster Linie auf die Unterstützung von ehrenamtlichen Helfern angewiesen. Diese Helfer haben jedoch oft andere Verpflichtungen wie Beruf, Studium oder Familie, die ihre Verfügbarkeit beeinflussen können. Die Herausforderung besteht darin, den Wachplan so zu gestalten, dass er die Zeiten der Helfer berücksichtigt und sicherstellt, dass zu jeder Zeit ausreichend Kräfte vor Ort sind.

**Qualifikationen und Ausbildungsstand:** Für die Wachplanerstellung ist es wichtig, die Qualifikationen und den Ausbildungsstand der Helfer zu berücksichtigen. Je nach Art der Wasserrettungseinsätze und Gewässerbedingungen können bestimmte Fähigkeiten und Zertifizierungen erforderlich sein. Es stellt sich als komplex heraus, den Wachplan so zu erstellen, dass qualifizierte Helfer an den richtigen Stellen eingesetzt werden und die Sicherheitsstandards gewährleistet sind.

**Flexibilität und Änderungen:** Der Wachplan muss flexibel genug sein, um Änderungen und unvorhergesehene Ereignisse zu berücksichtigen. Dies können beispielsweise kurzfristige Absagen von Helfern oder Änderungen in den Einsatzanforderungen sein. Der Wachplan muss so gestaltet werden, dass er leicht anpassbar ist und dennoch eine kontinuierliche Gewässerüberwachung gewährleistet.

**Kommunikation und Koordination:** Die Wachplanerstellung erfordert eine effektive Kommunikation und Koordination innerhalb der Wasserwacht. Informationen über Verfügbarkeiten, Änderungen und Einsatzanforderungen müssen schnell und zuverlässig zwischen den Verantwortlichen ausgetauscht werden.

Die Wachplanerstellung in der Wasserwacht ist eine komplexe Aufgabe, die sorgfältige Planung, Koordination und Berücksichtigung verschiedener Faktoren erfordert. Durch den Einsatz moderner Technologien wie einer Webanwendung können viele dieser Herausforderungen bewältigt und die Effizienz bei der Wachplanerstellung gesteigert werden. Eine gut konzipierte und durchdachte Lösung kann dazu beitragen, die Wachplanung zu optimieren und einen reibungslosen Ablauf der Gewässerüberwachung und Rettungseinsätze zu gewährleisten.

### 2.3. Bedeutung von IT-Lösungen zur Digitalisierung von Verwaltungsprozessen

Die Bedeutung von IT-Lösungen zur Digitalisierung von Verwaltungsprozessen ist in der heutigen Zeit von großer Relevanz. Traditionell wurden Verwaltungsprozesse in vielen Organisationen, einschließlich der Wasserwacht, manuell oder mit papierbasierten Systemen abgewickelt. Dies führte häufig zu ineffizienten Abläufen, einer hohen Fehleranfälligkeit und einem erheblichen Zeitaufwand.

Die Einführung von IT-Lösungen zur Digitalisierung von Verwaltungsprozessen bietet zahlreiche Vorteile. Ein zentraler Aspekt ist die Automatisierung von Aufgaben und die Reduzierung manueller Eingriffe. Durch den Einsatz von speziell entwickelten Webanwendungen oder Softwarelösungen können Verwaltungsprozesse optimiert und beschleunigt werden. Die erforderlichen Informationen können elektronisch erfasst, gespeichert und bearbeitet werden, was zu einer erheblichen Zeitersparnis führt.

Darüber hinaus ermöglichen IT-Lösungen eine verbesserte Datenverwaltung und -analyse. Durch die Digitalisierung von Verwaltungsprozessen können Daten effizient erfasst, organisiert und ausgewertet werden. Dies ermöglicht es der Wasserwacht, wertvolle Erkenntnisse zu gewinnen, Trends zu erkennen und fundierte Entscheidungen zu treffen. Beispielsweise können Einsatzstatistiken, Mitgliederdaten oder Materialbestände elektronisch erfasst und in Echtzeit analysiert werden, um einen umfassenden Überblick zu erhalten.

Des Weiteren führt die Digitalisierung von Verwaltungsprozessen zu einer verbesserten Kommunikation und Zusammenarbeit. Durch den Einsatz von IT-Lösungen können Informationen schnell und zuverlässig zwischen den relevanten Personen und Abteilungen ausgetauscht werden. Dies erleichtert die Koordination von Aufgaben, die Planung von Einsätzen und die effektive Zusammenarbeit. Durch den Zugriff auf gemeinsame Datenbanken oder Kollaborationstools können Informationen in Echtzeit geteilt werden, was zu einer erhöhten Effizienz und Transparenz führt.

Nicht zuletzt tragen IT-Lösungen zur Digitalisierung von Verwaltungsprozessen auch zur Verbesserung der Servicequalität bei. Durch den Einsatz von automatisierten Prozessen und einem benutzerfreundlichen Interface können Verwaltungsaufgaben effizienter erledigt werden. Dies ermöglicht es der Wasserwacht, ihre Ressourcen effektiver einzusetzen und sich auf ihre Kernaufgaben, wie die Wasserrettung und den Katastrophenschutz, zu konzentrieren. Gleichzeitig können die Mitglieder der Wasserwacht von einer vereinfachten Kommunikation und einer besseren Organisation profitieren.

Insgesamt ist die Bedeutung von IT-Lösungen zur Digitalisierung von Verwaltungsprozessen in der Wasserwacht nicht zu unterschätzen. Die Einführung solcher Lösungen ermöglicht eine effizientere Arbeitsweise, eine verbesserte Datenverwaltung und -analyse, eine gesteigerte Zusammenarbeit sowie eine höhere Servicequa-



lität. Die Digitalisierung von Verwaltungsprozessen bietet somit eine bedeutende Chance für die Wasserwacht, ihre Abläufe zu optimieren und ihre Leistungsfähigkeit zu steigern.

## 2.4. Best Practices Websitedesign

### 2.4.1. Usability

Ein Hauptmerkmal, auf das bei der Entwicklung von Webanwendungen Wert gelegt werden soll, ist Usability.

Jakob Nielsen definiert Usability als ein Qualitätsmerkmal, das bewertet, wie einfach Benutzeroberflächen zu bedienen sind. Usability umfasst fünf Qualitätskomponenten: Lernbarkeit (wie einfach es für Benutzer ist, grundlegende Aufgaben beim ersten Kontakt mit dem Design zu erledigen), Effizienz (wie schnell Benutzer Aufgaben erledigen können, nachdem sie das Design gelernt haben), Erinnerbarkeit (wie leicht Benutzer ihre Fertigkeiten wiedererlangen können, nachdem sie das Design eine Zeit lang nicht verwendet haben), Fehler (wie viele Fehler Benutzer machen, wie schwerwiegend diese sind und wie leicht sie sich davon erholen können) und Zufriedenheit (wie angenehm die Nutzung des Designs ist) (vgl. [Nie]).

### 2.4.2. Responsive Design

Im heutigen digitalen Zeitalter, in dem Benutzer eine Vielzahl von Geräten verwenden, um auf Websites zuzugreifen, ist das Konzept des Responsive Designs von großer Bedeutung. Responsive Design bezieht sich darauf, wie eine Website gestaltet wird, um sich an verschiedene Bildschirmgrößen anzupassen, einschließlich Desktop-Computern, Tablets und Smartphones (vgl. [AM17], S. 50).

Eine der grundlegenden Techniken des Responsive Designs ist die Verwendung von flüssigen Rastern. Statt fester Pixelmaße passen sich die Layouts automatisch an den verfügbaren Platz auf dem Bildschirm an. Dies ermöglicht eine konsistente Darstellung unabhängig von der Bildschirmgröße (vgl. [Kad12], Kapitel 2).

Mit Media Queries lassen sich CSS-Regeln formulieren, die auf verschiedene Bildschirmgrößen reagieren. Sie ermöglichen es, spezifische Stile basierend auf den Abmessungen des Geräts anzuwenden. Auf diese Weise kann das Design der Website für verschiedene Bildschirme optimiert werden (vgl. [Kad12], Kapitel 3).

Bilder und Medienelemente sollten so gestaltet sein, dass sie sich proportional zur Bildschirmgröße ändern. Die Verwendung von Bildern mit variabler Größe, die sich je nach Display anpassen, verhindert das Überladen von Bildern auf kleineren Bildschirmen (vgl. [Kad12], Kapitel 4).

## *2. Theoretischer Hintergrund*

Responsive Design trägt zur Benutzerfreundlichkeit bei, indem es sicherstellt, dass Benutzer die gleiche Qualität der Benutzererfahrung unabhängig vom Gerät genießen. Dies verhindert unangenehme Zoom- oder Scrollvorgänge auf kleinen Bildschirmen (vgl. [Cho]).

Zusammenfassend ist Responsive Design ein wesentlicher Bestandteil eines erfolgreichen Websitedesigns. Durch die Verwendung von flüssigen Rastern, flexiblen Bildern und Media Queries wird sichergestellt, dass eine Website auf verschiedenen Geräten optimal angezeigt wird. Dies verbessert die Benutzererfahrung, erhöht die Benutzerzufriedenheit und trägt zur Erreichung der Website-Ziele bei.

## 3. Anforderungsanalyse

### 3.1. Identifikation von Anforderungen an die Webanwendung

Bevor mit der eigentlichen Implementierung der Anwendung gestartet werden kann, müssen zunächst die Anforderungen daran festgemacht werden.

In einem initial bereitgestelltem Dokument (*siehe Anhang A.1*) sind die ersten Entwürfe der Anwendung aufgezeichnet. Darin sieht man mehrere gewünschte Features, um die Verwaltungsarbeiten der Wasserwacht abzudecken, wie beispielsweise „Wachplanung“, „Materialcheck“, „Wachbuch“ oder „Sanitätsbuch“. In dieser Arbeit werden hauptsächlich die Punkte „Wachplanung“ und „Wachbuch“ ausgearbeitet, welche im folgenden detaillierter beschrieben werden.

#### 3.1.1. Nutzergruppen

Für die Anwendung werden zwei Nutzergruppen angefordert.

Zum einen sind das „Leitungskräfte / Admins“. Diese sind für administrative Tätigkeiten zuständig und erledigen die Planung und Verwaltung der Anwendung. Außerdem gibt es noch die „Mitglieder / Helfer“. Diese können sich für einzelne Wachtage einbuchen und danach ausführen.

#### 3.1.2. Wachplanung

Leitungskräften soll es ermöglicht werden im Voraus Termine für die einzelnen Wachpläne einzustellen. Diese sollen für ein konkretes Datum und Uhrzeit erstellt werden können, oder als Serie mit einer bestimmten Laufzeit. In einem Kalender sollen die einzelnen Termine dargestellt werden, mit einer farblichen Kennung, ob für diesen Termin bereits Mitarbeiter eingebucht sind.

Für Mitglieder ist es notwendig, sich für Wachtage einzubuchen. Dazu soll auch der Kalender genutzt werden, um eine Übersicht der verfügbaren Wachtage zu erhalten.

#### 3.1.3. Wachbuch

Im Wachbuch werden die Ereignisse eines Wachtages festgehalten. Nutzern soll es möglich sein, eine Liste der gebuchten Helfern anzuzeigen, sowie nach Wachstart eine Liste der anwesenden Helfern anzuzeigen. Nutzer können sich für diesen Wachtage ein-, bzw. ausbuchen.

Weiterhin sollen für diesen Wachtage die vorliegenden Wetterdaten dargestellt werden, welche über eine API ermittelt werden. Nutzer können die aktuelle Wassertemperatur über ein Eingabefeld abspeichern.

Die Ereignisse des Wachtags werden in einer Tabelle mit Namen und Zeitstempel festgehalten. Dabei werden der Wachbeginn und das Wachende der Mitglieder, die An- und Abmeldung in der ILS (Integrierte Leitstelle), die automatischen Wetterdaten und der Wachstart/ -ende automatisch eingetragen. Nutzer können zusätzlich manuell Eingaben tätigen.

Leitungskräften soll es zusätzlich ermöglicht werden eine Auswertung eines Wachtages vorzunehmen. Dazu gibt es einen Button „Drucken“, über den ein PDF, welches die Ereignisse des Wachtages zusammenfasst, heruntergeladen werden kann. Die Informationen, welche dieses Dokument enthalten muss, wurden aus dem derzeit verwendeten Wachbuch übernommen (*siehe Anhang A.2*). Darin befinden sich die vorliegenden Wetterdaten, Informationen über das anwesende Wachpersonal, sowie das Wachtagebuch.

Die Punkte „Übersicht Checks“ und „Einsätze“ wurden im Rahmen dieser Arbeit aus zeitlichen Gründen nicht betrachtet.

## 3.2. Befragung von Wasserwachtmitgliedern und Mockups

In mehreren Iterationen wurden mit Herrn Andreas Schmeisl (Vorsitzender Kreiswasserwacht Dingolfing-Landau) und Herrn Werner Gerl (Technischer Leiter Kreiswasserwacht Dingolfing-Landau) die finalen Vorgaben geklärt. Dabei wurde mit Hilfe von Mockups ein Zielbild der Anwendung entwickelt.

### 3.2.1. Mockups

#### Was sind Mockups

Mockups werden in der Softwareentwicklung eingesetzt um sicher zu gehen, dass man die Anforderungen an die Anwendung verstanden hat. Mockups haben den Vorteil, dass sie kosten- und zeiteffizient erstellt werden können und schnell anpassbar sind.

#### Papier-Mockups

In einem initialen Meeting wurden mit Papier Mockups die groben Umrisse der Website aufgezeigt. Es wurden außerdem noch weitere Funktionalitäten und Anforderungen besprochen, die aus den vorliegenden Unterlagen nicht hervorgingen. Mit den Ergebnissen aus diesem Meeting konnte ein Prototyp der Anwendung erstellt werden.

#### **Figma**

Figma ist eine Website, über die man Prototypen und Mockups erstellen kann (sh. [fig]). Über einzelne Boards können die verschiedenen Seiten der Zielanwendung simuliert werden. Der Vorteil hierbei ist, dass auch Nutzereingaben simuliert werden können. So kann beispielsweise bei Klicken ein Seitensprung animiert werden. Dadurch kann man ohne viel Aufwand dem Kunden zeigen, wie die Anwendung einmal aussehen wird.

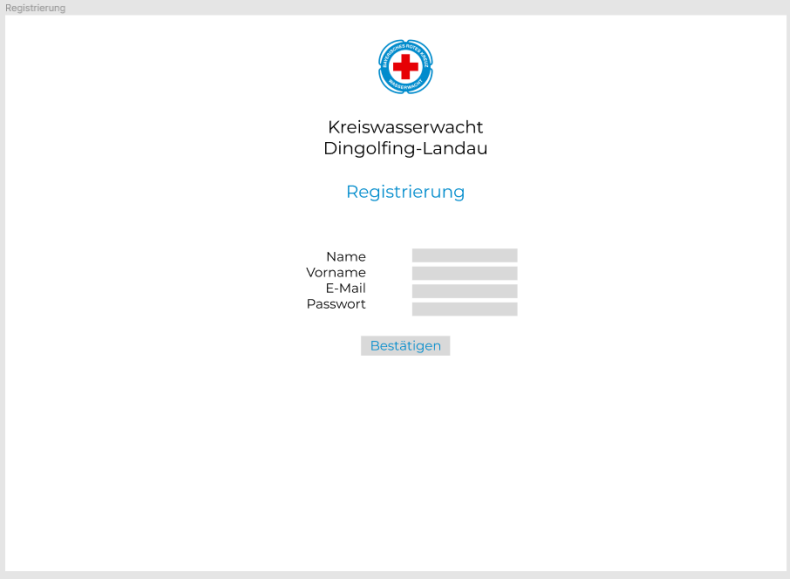
#### **3.2.2. Ergebnisse**

Mit Figma wurden alle vorgesehenen Seiten und Funktionalitäten der Website simuliert. Im folgenden werden die Ergebnisse und die weiteren besprochenen Anforderungen präsentiert.

#### **Anmeldeprozess**

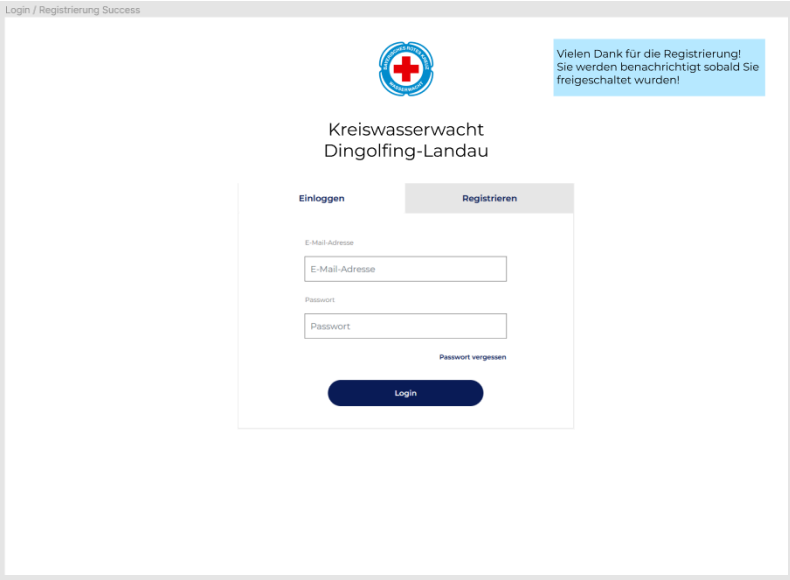
Beim Login Prozess wurde gefordert, dass die „Admins“ bei einer Neu-Anmeldung über Email benachrichtigt werden. Neue Nutzer können daher nach dem Registrieren nicht sofort auf die Seite zugreifen, sondern müssen erst warten bis ein Admin sie bestätigt hat. Dadurch soll sichergestellt werden, dass nur Wasserwachtmitglieder Zugang zum System erhalten.

### 3. Anforderungsanalyse



Registration form for Kreiswasserwacht Dingolfing-Landau. The form includes a logo at the top, the organization's name, and a heading for registration. Below this are input fields for Name, Vorname, E-Mail, and Passwort, followed by a 'Bestätigen' button.

(a) Registrierung



Login and registration success screen for Kreiswasserwacht Dingolfing-Landau. The screen displays a message: 'Vielen Dank für die Registrierung! Sie werden benachrichtigt sobald Sie freigeschaltet wurden!'. Below this is a login form with fields for E-Mail-Adresse and Passwort, a 'Passwort vergessen' link, and a 'Login' button. There are also tabs for 'Einloggen' and 'Registrieren'.

(b) Benachrichtigung nach Registrierung

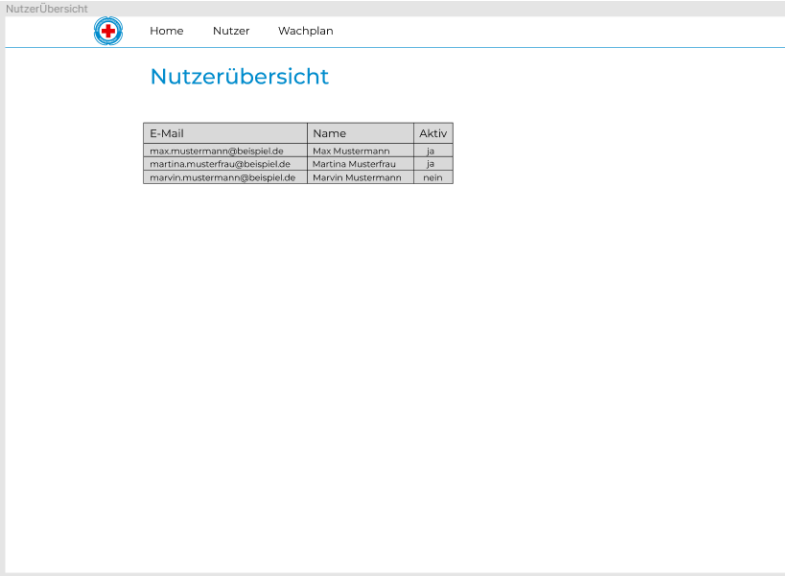
Abbildung 3.1.: AnmeldeProzess

#### Nutzerübersicht

Admins sehen auf der Nutzerübersicht alle Benutzer und ob sie bereits freigeschaltet wurden. Über das jeweilige Profil können sie die Person über einen Button freischalten. Danach wird derjenige per Email benachrichtigt und erhält Zugang zum System. Außerdem soll für jedes Mitglied eine Statistik erhoben werden, wie viele Stunden im Dienst verbracht wurden. Dies wurde als Anforderung mit aufgenommen, wurde jedoch im Mock nicht mehr eingearbeitet sondern direkt in die tatsächliche Anwen-

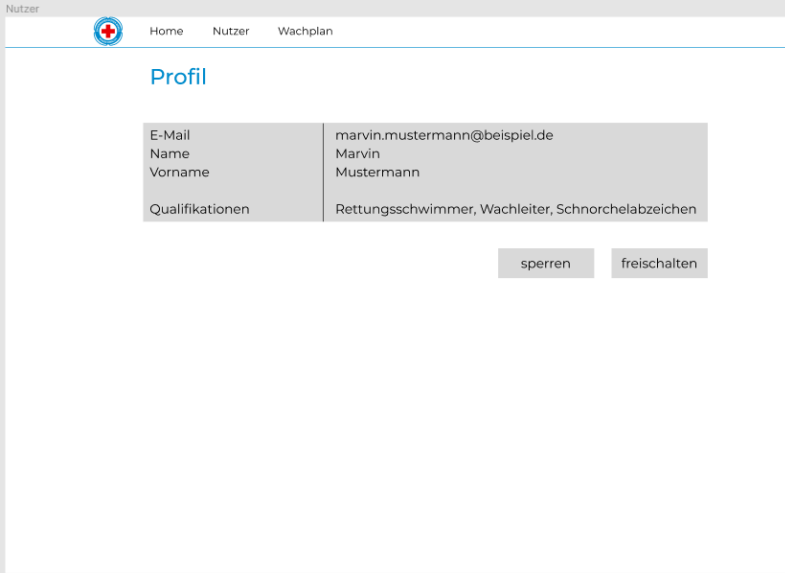
### 3. Anforderungsanalyse

dung eingebaut.



E-Mail	Name	Aktiv
max.mustermann@beispiel.de	Max Mustermann	ja
martina.mustermann@beispiel.de	Martina Mustermann	ja
marvin.mustermann@beispiel.de	Marvin Mustermann	nein

(a) Nutzerübersicht



E-Mail	marvin.mustermann@beispiel.de
Name	Marvin
Vorname	Mustermann
Qualifikationen	Rettungsschwimmer, Wachleiter, Schnorchelabzeichen

(b) Profil Admin Ansicht

Abbildung 3.2.: Freischaltungs-Prozess

### Wachplanung

Bei der Wachplanung wurde angefordert, dass Wachtermine einzeln für ein bestimmtes Datum und Uhrzeit angelegt werden können. Außerdem soll es möglich sein gleich eine Reihe an Terminen anzulegen. Dazu muss ein Start- und Enddatum angegeben werden, die Uhrzeit, sowie die Wochentage an denen ein Termin eingestellt werden

### *3. Anforderungsanalyse*


soll.

Als Übersicht für die bisherigen Termine wird ein Kalender verwendet. Hier wurde überlegt, ob man die einzelnen Termine farbig codiert, wenn beispielsweise ein „Rettungsschwimmer“ und „Wachleiter“ noch nicht eingebucht sind. In weiteren Abstimmungsterminen hat man sich aber darauf geeinigt, die Termine rot einzufärben, wenn noch niemand eingebucht ist, und grün einzufärben wenn Nutzer eingebucht sind.



### 3. Anforderungsanalyse

Wachplan anlegen


Home
Nutzer
Wachplan

## Wachplan anlegen

Einzeltermin anlegen
Terminreihe anlegen

Datum

Wachbeginn

Wachende

speichern

(a) Einzeltermin anlegen

Wachplan anlegen

[Home](#) [Nutzer](#) [Wachplan](#)

## Wachplan anlegen

Einzeltermin anlegen

Terminreihe anlegen

Start-Datum

24.05.2022

End-Datum

27.05.2022

Wachbeginn

10:00

Wachende

17:00

Montag

Dienstag

Mittwoch

Donnerstag

Freitag

Samstag

Sonntag

speichern

(b) Terminreihe anlegen

Wachplan

Home Nutzer Wachplan

## Wachplan

23.05.2022

	23.05.2022	24.05.2022	25.05.2022	26.05.2022	27.05.2022	28.05.2022	29.05.2022
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							

Termine anlagen

(c) Wachplan Übersicht

#### **Wachplandurchführung**

Über einen Klick auf die Terminelemente im Wachplan kommt man auf die Durchführungsseite des jeweiligen Wachtages. Diese Seite wurde entsprechend der initialen Vorgabe (*siehe Anhang A.1*) nachmodelliert.

Zunächst findet man oben auf der Seite das Tagesdatum, sowie die aktuelle Uhrzeit. Daneben findet man die Uhrzeit des Wachbeginns und Wachendes.

Rechts oben befindet sich die Ausgabe der aktuellen Wetterdaten, welche automatisch eingespielt werden sollen. Hierfür wurde nach einer passenden Wetter-Anwendung gesucht, welche eine API zur Abfrage der Daten bereitstellt. Darunter befindet sich ein Eingabefeld für die aktuelle Wassertemperatur. Diese wird von Wasserwachtmitgliedern gemessen und im System hinterlegt.

Nachfolgend befinden sich die Tabellen mit den Helfern für den Wachttag. Auf der linken Seite befindet sich die Tabelle mit den gebuchten Helfern. Ist ein Wachttag noch nicht gestartet, können sich Nutzer über ein darunter liegendes Eingabefeld einbuchen. Somit kann eine Planung der jeweiligen Wachtage stattfinden und man erhält einen Überblick, wie viele Personen planen vor Ort zu sein. Rechts daneben befindet sich die Tabelle der anwesenden Helfer. Sobald ein Wachtag gestartet wurde können sich die Nutzer über das selbe Eingabefeld einbuchen und ihren Wachbeginn dokumentieren. Hier wurde in weiteren Besprechungen noch angefordert, dass es möglich sein soll Nutzer aus einer Auswahlliste auszuwählen und einzubuchen und zusätzlich auch über ein Freitextfeld einzugeben. Die Möglichkeit Nutzer über einen Freitext hinzufügen zu können wird benötigt, da auch Helfer am Dienst teilnehmen können, die nicht im System als Nutzer registriert sind.

Über den Button unterhalb kann eine An- und Abmeldung in der ILS dokumentiert werden. Daneben befindet sich der Button zum starten und beenden eines Wachtages.

Darunter befindet sich ein größeres Texteingabefeld, wodurch Geschehnisse während des Wachtages im Wachbuch festgehalten werden können.

Das Wachbuch enthält alle Ereignisse eines Wachtages. Ein Eintrag enthält immer einen Zeitstempel und den jeweiligen Nutzer der den Eintrag verfasst hat, bzw „System“ wenn der Eintrag automatisch eingefügt wurde. Das Wachbuch dokumentiert somit Wachbeginn und Ende, Wachbeginn und Ende der jeweiligen Nutzer, die Wetterdaten, die Wassertemperatur, An- und Abmeldungen im ILS, sowie eigene Ereignisangaben der Nutzer.

Ein Feature welches im Mock noch nicht dargestellt wurde ist das Drucken eines Wachtages. Nach Beendigung eines Wachtages soll es möglich sein die gesammelten Daten als PDF herunterzuladen. Dafür wurde ein Dokument bereitgestellt, welches aktuell benutzt wird um den Wachtag zu protokollieren. Nach dieser Vorlage soll auch das PDF erstellt werden mit den Daten aus dem System.

Wachbuch

Home
Nutzer
Wachplan

## Wachplan

23.05.2022  
09:45
Wachbeginn: 10:00  
Wachende: 17:00

25 °C | °F
Niederschlag: 10%  
Luftfeuchte: 53%  
Wind: 16 km/h

Wassertemperatur:  ✓

Gebuchte Helfer

Max Mustermann  
Martina Musterfrau  
Merlin Mustermann

Anwesende Helfer

Ereignis...

Datum / Zeit	Name	Beschreibung

(a) Wachstart

Wachbuch4

Home
Nutzer
Wachplan

## Wachplan

23.05.2022  
17:03
Wachbeginn: 10:00  
Wachende: 17:00

25 °C | °F
Niederschlag: 10%  
Luftfeuchte: 53%  
Wind: 16 km/h

Wassertemperatur:  ✓

Gebuchte Helfer

Max Mustermann  
Martina Musterfrau  
Merlin Mustermann  
Marvin Mustermann

Anwesende Helfer

Marvin Mustermann  
Martina Musterfrau  
Max Mustermann

Ereignis...

Datum / Zeit	Name	Beschreibung
23.05.2022 / 17:03	Marvin Mustermann	Wachende
23.05.2022 / 16:00	Marvin Mustermann	Wetter: Sonnig, 28 °C
23.05.2022 / 15:45	Martina Mustermann	Wassertemperatur 20 °C
23.05.2022 / 12:30	Marvin Mustermann	Einsatz: Rettung vor Ertrinken
23.05.2022 / 11:30	Martina Mustermann	Wassertemperatur 19,5 °C
23.05.2022 / 10:45	Max Mustermann	Abmeldung ILS
23.05.2022 / 10:30	Max Mustermann	Anmeldung ILS
23.05.2022 / 09:45	Marvin Mustermann	Wachbeginn

(b) Wachende

Abbildung 3.4.: Wachbuch

## 4. Technologien und Methoden

### 4.1. Auswahl geeigneter Technologien und Tools für die Webanwendung

Bei der Auswahl der Technologien und Tools für die Entwicklung meiner Webanwendung habe ich mich bewusst für Tools entschieden, mit denen ich bereits praktische Erfahrung sammeln konnte. Während meines Entwicklungspraktikums an der Universität habe ich mit den gleichen Tools gearbeitet, nämlich Java als Programmiersprache, das Spring-Framework für die Backend-Entwicklung, Thymeleaf für die serverseitige HTML-Vorlagenverarbeitung und Bootstrap für das Frontend-Design. Diese praktische Erfahrung ermöglichte mir einen reibungslosen Start in das Projekt, da ich bereits mit den Konzepten, Funktionalitäten und bewährten Methoden dieser Tools vertraut war. Durch die Verwendung von Java und den genannten Frameworks konnte ich eine solide und effiziente Entwicklungsumgebung schaffen, um die Anforderungen der Webanwendung erfolgreich umzusetzen. Im Folgenden werden die angewandten Technologien näher beschrieben.

### 4.2. Beschreibung der verwendeten Programmiersprachen, Frameworks und Datenbanken

#### 4.2.1. Java

Java ist eine vielseitige und populäre Programmiersprache, die von Anfängern bis hin zu Profis genutzt wird. Sie zeichnet sich durch Einfachheit und Flexibilität aus und ermöglicht die Entwicklung komplexer Anwendungen.

In Webanwendungen bietet Java mehrere Vorteile. Dank der objektorientierten Natur von Java können Entwickler einen wiederverwendbaren Code erstellen, was die Entwicklung und Wartung von Webanwendungen erleichtert. Java-Anwendungen sind plattformunabhängig, da sie auf einer virtuellen Maschine (JVM) laufen, was die Kompatibilität mit verschiedenen Betriebssystemen gewährleistet (vgl. [sca]).

Zusätzlich profitieren Entwickler von der breiten Unterstützung, einer großen Entwicklergemeinschaft und einer reichhaltigen Auswahl an Ressourcen (vgl. [sca]). Dies trägt dazu bei, effiziente und erfolgreiche Webanwendungen zu erstellen, die den Anforderungen der modernen Online-Welt gerecht werden.

### 4.2.2. Spring

Spring ist ein Open-Source Java Framework und bildet die Grundlage für dieses Projekt. Das Spring Framework wurde entwickelt um die Erstellung von Anwendungen zu erleichtern.

Kernfunktionalität von Spring ist die Inversion of Control (IoC) und die Dependency Injection (DI), die es ermöglicht, Abhängigkeiten zwischen Objekten zu verwalten. Durch die IoC kann eine Komponente Abhängigkeiten zu anderen Objekten definieren, ohne selbst diese Objekte instanziierten zu müssen. Der IoC Container fügt diese Abhängigkeiten dem Objekt ein (DI) (vgl. [\[spra\]](#)).

Das Spring-Framework deckt verschiedene Bereiche ab, darunter Webentwicklung, Datenbankzugriff, Sicherheit, Messaging und mehr. Spring Boot, eine Erweiterung von Spring, vereinfacht die Erstellung von Stand-alone-Anwendungen und Mikroservices, indem es eine vorkonfigurierte Umgebung bereitstellt (vgl. [\[sprb\]](#)).

Ein weiterer Vorteil von Spring ist die große Community die dahinter steht. Für die verschiedenen Spring-Tools werden zahlreiche Tutorials und Dokumentation bereitgestellt, die eine große Hilfe für den Einstieg darstellen.

Insgesamt erleichtert das Spring-Framework die Entwicklung von Java-Anwendungen erheblich, indem es bewährte Designmuster, Best Practices und Abstraktionen zur Verfügung stellt. Es ermöglicht es Entwicklern, sich auf die eigentlichen Funktionalitäten ihrer Anwendungen zu konzentrieren und gleichzeitig eine flexible, skalierbare und wartbare Architektur aufzubauen.

### 4.2.3. Thymeleaf

Thymeleaf ist eine leistungsstarke serverseitige Template-Engine für die Erstellung von Webseiten in Java-basierten Webanwendungen. Sie ermöglicht die nahtlose Integration von dynamischen Inhalten in HTML-Dokumente und erleichtert die Entwicklung von ansprechenden und interaktiven Benutzeroberflächen.

Ein herausragendes Merkmal von Thymeleaf ist seine Fähigkeit, direkt in HTML-Dokumenten zu arbeiten, wodurch die Integration von Serverdaten nahtlos erfolgen kann. Durch die Verwendung von speziellen Attributen und Tags in HTML-Dateien können Entwickler serverseitige Daten in die Darstellung einbetten und komplexe UI-Elemente erstellen.

Thymeleaf unterstützt verschiedene Template-Layouts, Fragmente und Iterationen, die die Wiederverwendbarkeit von Code fördern. Dies ermöglicht eine effiziente Entwicklung von konsistenten und wartbaren Benutzeroberflächen (vgl. [\[thyb\]](#)).

Ein weiterer Vorteil von Thymeleaf ist seine Integration mit Spring. Es passt nahtlos in das Spring-Framework und ermöglicht die einfache Verbindung von Daten aus dem Backend mit den Frontend-Templates. Dies erleichtert die Entwicklung von dynamischen Webseiten, ohne die Trennung zwischen Backend und Frontend zu beeinträchtigen (vgl. [\[thya\]](#)).

Insgesamt bietet Thymeleaf eine effektive Möglichkeit, serverseitige Daten in HTML-Dateien zu integrieren und interaktive Webseiten zu erstellen. Es ist besonders gut in Kombination mit dem Spring-Framework geeignet und trägt zur Entwicklung von modernen und funktionalen Webanwendungen bei.

### 4.2.4. Bootstrap

Bootstrap ist ein Open-Source-Framework für das Frontend-Webdesign und die Entwicklung von responsiven und benutzerfreundlichen Benutzeroberflächen. Es wurde von Twitter entwickelt und stellt eine Sammlung von vorgefertigten HTML-, CSS- und JavaScript-Komponenten zur Verfügung, die Entwicklern helfen, schnell ansprechende Webseiten zu erstellen (vgl. [Ott]).

Ein herausragendes Merkmal von Bootstrap ist seine Fähigkeit zur Erstellung responsiver Designs. Das bedeutet, dass Webseiten, die mit Bootstrap erstellt werden, sich automatisch an verschiedene Bildschirmgrößen und Geräte anpassen können, darunter Desktops, Tablets und Smartphones. Dies trägt zur Verbesserung der Benutzererfahrung auf verschiedenen Plattformen bei (vgl. [Spu13], S. 7).

Bootstrap bietet eine breite Palette von wiederverwendbaren UI-Komponenten wie Navigationsleisten, Buttons, Formularen, Karten, Modals und vieles mehr. Diese Komponenten können einfach in Webseiten integriert werden, wodurch die Entwicklung beschleunigt und konsistente Designs gewährleistet werden. Das Framework enthält auch ein flexibles Rastersystem, das es Entwicklern ermöglicht, das Layout einer Webseite in Spalten und Zeilen zu strukturieren. Dies erleichtert die Erstellung komplexer Layouts und die Ausrichtung von Inhalten (vgl. [booa]).

Bootstrap bietet auch vorgefertigte CSS-Klassen und JavaScript-Funktionalitäten, um häufig verwendete Funktionen wie Animationen, Navigation und Modals zu implementieren (vgl. [boob]). Dies ermöglicht es Entwicklern, interaktive Elemente ohne umfangreichen individuellen Code hinzuzufügen.

Insgesamt erleichtert Bootstrap die Entwicklung moderner Webseiten erheblich, indem es Entwicklern ermöglicht, schnell hochwertige und responsiv gestaltete Benutzeroberflächen zu erstellen. Es ist besonders nützlich für Projekte, die ein ansprechendes Design und eine konsistente Benutzererfahrung erfordern.

### 4.2.5. Postgres & Spring Data JPA

In meiner Bachelorarbeit habe ich PostgreSQL als Datenbankmanagementsystem (DBMS) verwendet, um die Daten meiner Webanwendung effizient zu speichern. Die Entscheidung für PostgreSQL als DBMS basierte auf seinen bewährten Funktionen für Datenpersistenz und -verwaltung (vgl. [pos]), sowie auf seiner nahtlosen Integration mit dem verwendeten Spring Data JPA-Framework (vgl. [sprc]).

PostgreSQL bietet eine solide Grundlage für die Datenpersistenz. In meiner Anwendung habe ich Daten in Form von Entitäten gespeichert. Entitäten sind Java-Klassen, die mit der Datenbanktabelle übereinstimmen und von Spring Data JPA verwaltet werden. Die Interaktion mit der Datenbank erfolgt durch objektorientierte Operationen, die von Spring Data JPA bereitgestellt werden (vgl. [sprd]). Dies ermöglicht es, Datenbankabfragen und -manipulationen auf einer höheren Abstraktionsebene zu handhaben.

Obwohl PostgreSQL eine breite Palette von erweiterten Funktionen bietet, die von räumlichen Abfragen bis zur Unterstützung komplexer Datentypen reichen (vgl. [pos]), habe ich mich auf die grundlegenden Aspekte der Datenpersistenz beschränkt. Dazu gehörten die Erstellung von Datenbanktabellen basierend auf meinen Entitäten, die Definition von Primärschlüsseln und Fremdschlüsselbeziehungen für die Datenintegrität, sowie die Verwendung von Indizes zur Optimierung von Abfragen.

Insgesamt ermöglichte die Kombination von PostgreSQL und Spring Data JPA eine effiziente und zuverlässige Datenpersistenz in meiner Webanwendung. Die solide Leistung von PostgreSQL und die praktische Abstraktion von Spring Data JPA erleichterten die Entwicklung und Wartung meiner Anwendung erheblich.

### 4.2.6. Wetter API

Um für einen Wochtag die aktuellen Wetter Daten speichern zu können, wurde eine Wetter-API integriert.

Die Integration der Wetter-API erforderte die Auswahl einer geeigneten API und die Einbindung der API-Anfragen in den Entwicklungsprozess. In meiner Bachelorarbeit habe ich mich für die API von VisualCrossing entschieden (s.h. [wea]), da sie zuverlässige und detaillierte Wetterdaten bereitstellt, die speziell auf den Standort der Wasserwacht zugeschnitten sind. Ein weiterer Vorteil dieser API ist, dass sie pro Tag 1000 Abfragen kostenfrei erlaubt.

Die Wetter-API ermöglichte es meiner Webanwendung, aktuelle Wetterdaten wie Temperatur, Luftfeuchtigkeit, Windgeschwindigkeit abzurufen. Diese wurden dann in die Benutzeroberfläche meiner Anwendung integriert, um den Benutzern leicht verständliche Informationen zur Verfügung zu stellen. Zusätzlich werden diese Daten auch automatisch in das Wachbuch geschrieben. So kann man auch nach Ende eines Wochtages die genauen Wetterverhältnisse nachvollziehen.

### 4.3. Agile Entwicklungsmethoden zur Umsetzung des Projekts

Die agile Umsetzungsmethode bildete das Rückgrat der Entwicklung meiner Webanwendung für die lokale Wasserwacht. In Anbetracht der sich ständig ändernden Anforderungen und der Notwendigkeit, schnell auf Feedback zu reagieren, erwies sich die Agile Methodik als äußerst geeignet, um Flexibilität und Effizienz in den Entwicklungsprozess zu integrieren.

In meiner Bachelorarbeit habe ich eine agile Herangehensweise gewählt, um die Entwicklung meiner Webanwendung zu steuern. Während ich mich von den Kernprinzipien agiler Methoden inspirieren ließ, habe ich die traditionellen Sprints mit festgelegten Zeiträumen nicht strikt umgesetzt. Stattdessen habe ich mich für eine iterative Vorgehensweise entschieden, bei der ich in einer bestimmten Zeitspanne so weit wie möglich gearbeitet habe und dann die Ergebnisse vorgestellt und diskutiert habe.

Die regelmäßige Abstimmung mit den Mitgliedern der Wasserwacht ermöglichte es schnell auf neue Anforderungen und Feedback zu reagieren. Gleichzeitig konnte ich dadurch den Fokus auf eine kontinuierliche Weiterentwicklung und Verbesserung der Anwendung legen.



## 5. Konzeption und Umsetzung

### 5.1. Detaillierte Beschreibung der Konzeption der Webanwendung

#### 5.1.1. Datenmodell

Für die Webanwendung ist es erforderlich die benötigte Datenmenge in einer Datenbank zu persistieren. Die einzelnen Entitäten habe ich in einem UML-Diagramm aufgezeichnet, um die Abhängigkeiten zwischen ihnen besser zu veranschaulichen. Anhand diesem werde ich im Folgenden das zu Grunde liegende Datenmodell beschreiben.

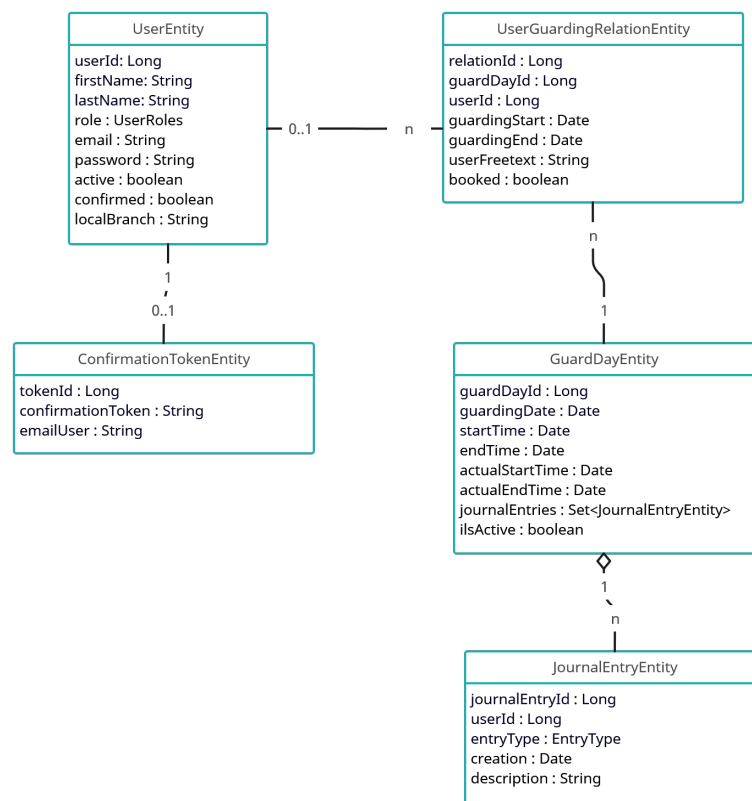


Abbildung 5.1.: UML-Diagramm

Zunächst kann festgestellt werden, dass jede Entität eine eindeutige Id besitzt. Diese dienen als Primärschlüssel der Entitäten, um eine eindeutige Identifikation zu ermöglichen.

### UserEntity

Mit der „UserEntity“ werden die Nutzerdaten repräsentiert. Zu Jedem Nutzer werden einige Kerndaten gespeichert, wie Vorname („firstName“), Nachname („lastName“), Email („email“) und Passwort („password“). Diese Daten dienen der Identifikation und Authentifizierung der Benutzer.

Die zugehörige Rolle eines Nutzers wird über das Attribut „role“ repräsentiert. Dahinter steckt die Enumeration „UserRoles“ mit den zugehörigen Werten „ROLE\_USER“ und „ROLE\_ADMIN“. „ROLE\_USER“ wird für einfache „Mitglieder / Helfer“ verwendet, während „ROLE\_ADMIN“ für „Leitungskräfte / Admins“ verwendet wird. Mit der Rolle eines Benutzers werden Berechtigungen innerhalb der Anwendung gesteuert und Zugriffsschichten festgelegt.

Das Feld „active“ spiegelt wieder, ob ein Anwender seine eigene Email-Adresse bei der Registrierung bestätigt hat. Das Feld „confirmed“ gibt wieder, ob der Nutzer durch einen Administrator freigeschaltet wurde. Wenn diese beiden Felder „true“ sind, erhält der Nutzer Zugang zur Anwendung. Dadurch wird sichergestellt, dass keine unbefugten Personen Zugang zur Anwendung erhalten.

Über das Feld „localBranch“ wird der jeweilige Ortsverband des Nutzers gespeichert.

### GuardDayEntity

Über die „GuardDayEntity“ werden die einzelnen Wachtage repräsentiert. Um einen Wachtag zu planen, braucht man ein Datum („guardingDate“), einen geplanten Startzeitpunkt („startingTime“) und einen geplanten Endzeitpunkt („endingTime“). Wenn ein Wachtag dann tatsächlich gestartet wird, wird nochmal der genaue Startzeitpunkt gemessen und gespeichert („actualStartTime“). Gleiches passiert auch beim Beenden mit dem Endzeitpunkt („actualEndTime“). Über diese Zeitpunkte lässt sich auch der Status eines Wachtages bestimmen, also ob er noch in Planung ist, ob er gerade in der Durchführung ist, oder ob er bereits beendet wurde.

In „journalEntries“ wird eine Liste der einzelnen Einträge im Wachbuch gespeichert. Dadurch sind die Einträge eindeutig einem Wachtage zugeordnet. Im Folgenden werden die „JournalEntryEntities“, die Bestandteile der „journalEntries“ sind, genauer beschrieben.

Über das Feld „isActive“ wird gespeichert ob gerade eine Verbindung zur integrierten Leitstelle vorhanden ist.

### JournalEntryEntity

In der „JournalEntryEntity“ werden die Einträge des Wachbuches festgehalten. Dazu wird einmal die „userId“ gespeichert, die den Nutzer repräsentiert, der den Eintrag verfasst hat. Wird keine „userId“ zu einem Eintrag gespeichert, kommt dieser Eintrag vom System selbst. Das passiert bei der automatischen Speicherung der Wetterdaten.

Jeder Eintrag hat einen „EntryType“, der angibt, um welche Art von Eintrag es sich handelt. Die verschiedenen Werte für „EntryType“ sind in Tabelle 5.1 aufgelistet. Dieser Typ erleichtert die Strukturierung und Kategorisierung der Einträge. Später werden bestimmte Kategorien gruppiert und im ausdrucksfähigen Wachbericht gesammelt dargestellt.

Außerdem werden pro Eintrag das Erstellungsdatum („creation“) und der eigentliche Inhalt („description“) gespeichert. So kann das Wachbuch chronologisch aufgebaut werden und ermöglicht es, die Ereignisse klar und detailliert zu dokumentieren.

### **UserGuardingRelationEntity**

Damit sich Helfer bei einem Wachtag einbuchen können, ist es notwendig eine entsprechende Relation zwischen den beiden Entitäten herzustellen, die diesen Zustand speichern kann. Dazu wurde die „UserGuardingRelationEntity“ modelliert. Sie enthält zunächst die „userId“ des Nutzers und die „guardDayId“ des jeweiligen Wachtages. Dadurch kann ein Nutzer eindeutig einem Wachtag zugeordnet werden. Wenn sich ein Nutzer für einen Wachtag anmeldet, wird eine „UserGuardingRelationEntity“ angelegt mit dem Attribut „booked“. Dies gibt an, dass sich ein Nutzer für diesen Wachtag einbucht und plant den Wachdienst an diesem Datum zu übernehmen.

Wenn ein Mitglied den Wachdienst antritt, wird eine neue Entität erstellt, um diesen Zustand zu dokumentieren. Diese Entität enthält das Attribut „booked“, das auf „false“ gesetzt wird, um anzuzeigen, dass der Nutzer den Wachdienst angetreten hat. Ebenso werden der Wachbeginn („guardingStart“) und das Wachende („guardingEnd“) in dieser neuen Entität festgehalten. Dies ermöglicht die Unterscheidung zwischen den Nutzern, die sich für den Wachdienst eingebucht haben, und denjenigen, die tatsächlich anwesend sind. Die ursprüngliche „UserGuardingRelationEntity“, die die Einbuchung des Nutzers dokumentiert, bleibt dabei unverändert.

Da es auch vorkommen kann, dass Helfer die Wasserwacht unterstützen, die nicht im System hinterlegt sind, besteht die Möglichkeit diesen per Freitext zu erfassen („userFreitext“). In diesem Fall bleibt das Attribut „userId“ leer.

### **ConfirmationTokenEntity**

Die „ConfirmationTokenEntity“ wird verwendet, um den Bestätigungsprozess für die Registrierung neuer Nutzer zu verwalten. Bei der Registrierung eines neuen Nutzers wird ein eindeutiger Bestätigungstoken („confirmationToken“) generiert und dem Nutzer per E-Mail als Bestätigungslink zugesandt. Die verwendete Email wird dabei auch in der Entität gespeichert im Attribut „emailUser“. Der Bestätigungstoken dient dazu, die Echtheit der E-Mail-Adresse des Nutzers sicherzustellen und sicherzustellen, dass die Registrierung vom Inhaber der E-Mail-Adresse initiiert wurde.

Die „ConfirmationTokenEntity“ spielt eine entscheidende Rolle bei der Sicherstellung der Authentizität der E-Mail-Adresse eines neuen Nutzers und gewährleistet

Tabelle 5.1.: Enumerationswerte für „EntryType“

Enumeration Wert	Bezeichnung
DEFAULT	Manuelle Meldung
WEATHER	Wetterbericht
WATER_TEMP	Wassertemperatur
USER_GUARD_BEGIN	Wachanmeldung Nutzer
USER_GUARD_END	Wachabmeldung Nutzer
GUARD_BEGIN	Wachbeginn
GUARD_END	Wachende
ILS_ACTIVE	Anmeldung ILS
ILS_INACTIVE	Abmeldung ILS

somit die Integrität des Registrierungsprozesses.

## 5.2. Funktionalitäten und Features der Anwendung

Im Folgenden werden die einzelnen Masken der fertigen Webanwendung präsentiert und erklärt.

### 5.2.1. Spring Security

Mithilfe von Spring Security kann festgelegt werden, welchen Nutzern welche Bereiche der Webanwendung zur Verfügung stehen. Das ist notwendig um sicherzustellen, dass nur Leitungskräften administrative Funktionalitäten angeboten werden. Dazu muss eine „Security Configuration“ angelegt werden, innerhalb der man den verschiedenen Rollen verschiedene Berechtigungen erteilen kann.

```

@Override
protected void configure(HttpSecurity httpSecurity) throws Exception{
    httpSecurity
        .authorizeRequests()
            .antMatchers("/users", "/guardday_creation").hasRole("ADMIN")
            .antMatchers("/profile", "/guardday_overview", "/guardday_execution/**").authenticated()
        .and()
            .authorizeRequests()
                .antMatchers("/resources/**", "/registration").permitAll()
                .anyRequest().permitAll()
        .and()
            .formLogin()
                .loginPage("/login")
                .failureUrl("/login-error")
                .defaultSuccessUrl("/index", true)
                .permitAll()
        .and()
            .logout()
                .permitAll();
}

```

Abbildung 5.2.: Auszug aus SecurityConfiguration.java

In diesem Fall wurde festgelegt, dass nur Admins auf die Bereiche „Nutzerübersicht“ und „Wachtagerstellung“ zugreifen können. Die Bereiche „Profil“, „Wachtagübersicht“ und „Wachtagdurchführung“ können hingegen von jedem authentifizierten Anwender aufgerufen werden.

Weiterhin kann auf der Controller Ebene der Zugriff auf einzelne Request Mappings beschränkt werden mit der Annotation „PreAuthorize“.

```
48     @PreAuthorize("#query == principal.getUserId().toString() ||" +  
49         "hasRole('ROLE_ADMIN')")  
50     @GetMapping(value = "/user/{query}")  
51     public String searchUser(@PathVariable("query") String query, Model model) {
```

Abbildung 5.3.: Auszug aus UserController.java

Im Beispiel 5.3 wird der Zugriff auf ein bestimmtes Profil nur gestattet, wenn der ausführende Nutzer die Rolle Admin hat („hasRole('ROLE\_ADMIN')“), oder der Anwender sein eigenes Profil sehen möchte („query == principal.getUserId().toString()“). Über Principal kann auf den gerade authentifizierte Nutzer zugegriffen werden.

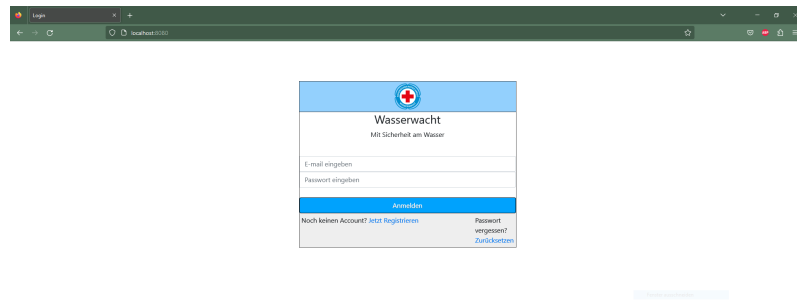
### 5.2.2. Anmelde- / Registrierungsprozess

Wenn man die Webanwendung das erste mal aufruft wird man auf die Login Seite geleitet. Besitzen die Helfer noch kein Konto, können sie sich über „Registrieren“ einen neuen Account erstellen. Dazu geben sie ihre Email an, legen ein Passwort fest und erfassen die Daten zur Person. Durch Klick auf „Registrieren“ wird ein neuer User angelegt und in der Datenbank gespeichert. In diesem Zustand kann der Nutzer die Website allerdings noch nicht betreten.

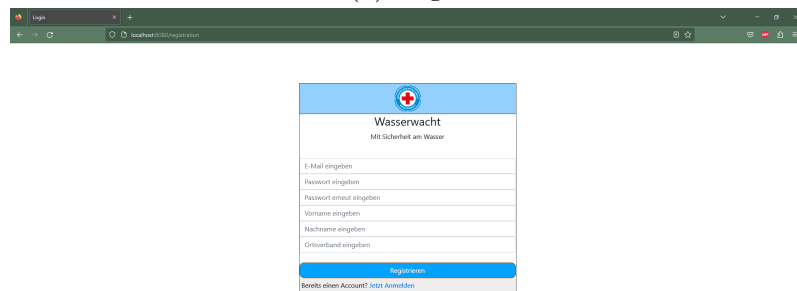
Im Hintergrund wird ein Bestätigungstoken generiert, der dem Anwender in Form eines Bestätigungslinks an die angegebene Email-Adresse versendet wird. Wenn man den Link aufruft, wird von der Anwendung überprüft, ob der erhaltene Bestätigungstoken mit dem in der Datenbank gespeicherten Token übereinstimmt. Wenn die Übereinstimmung erfolgreich ist, wird die Email-Adresse des Nutzers bestätigt. Somit wird sichergestellt, dass die Email-Adresse korrekt ist.

Parallel dazu wird an die hinterlegte Email-Adresse der Administratoren eine Email geschickt, um sie über eine neue Registrierung zu informieren. Sie müssen sich daraufhin im System anmelden und den Nutzer freischalten (siehe 5.7b).

## 5. Konzeption und Umsetzung



(a) Login



(b) Registrierung

Abbildung 5.4.: AnmeldeProzess

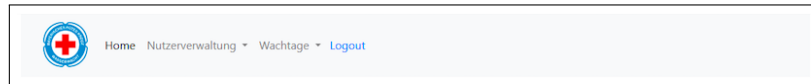
### 5.2.3. Navigationsleiste

Damit die Rollen „User“ und „Admin“ auch nur die Funktionalitäten angeboten kriegen, die sie verwenden dürfen, wurde die Navigationsleiste rollenspezifisch gestaltet (siehe 5.5). Normale Mitglieder haben über die Navigationsleiste nur Zugang zu ihrem eigenen Profil und zur Wachtagsübersicht, während Administratoren zusätzlich noch Zugang zur Nutzerübersicht und zur Wachtagerstellung haben.

## 5. Konzeption und Umsetzung



(a) Navigationsleiste - Nutzeransicht



(b) Navigationsleiste - Adminansicht

Abbildung 5.5.: Navigationsleisten

Dies konnte mit Hilfe von Thymeleaf's Integration von Spring Security umgesetzt werden. Über das „sec:authorize“ Attribut kann festgelegt werden, wann ein Element sichtbar ist. Im Beispiel 5.6 wird also der Menüpunkt „Profil“ nur den normalen Nutzern angezeigt, während den Admins eine Liste „Nutzerverwaltung“ mit den Unterpunkten „Profil“ und „Nutzerübersicht“ angezeigt wird.

```
<li class="nav-item" sec:authorize="hasRole('USER')" th:if="${#authorization.expression('isAuthenticated()')}">
  <a class="nav-link" th:href="@{/profile}">Profil</a>
</li>
<li class="nav-item dropdown" sec:authorize="hasRole('ADMIN')" th:if="${#authorization.expression('isAuthenticated()')}">
  <a class="nav-link dropdown-toggle" href="#" id="dropdownMenuUser" data-toggle="dropdown">
    Nutzerverwaltung
  </a>
  <ul class="dropdown-menu">
    <li><a class="dropdown-item" th:href="@{/profile}">Profil</a></li>
    <li><a class="dropdown-item" th:href="@{/users}">Nutzerübersicht</a></li>
  </ul>
</li>
```

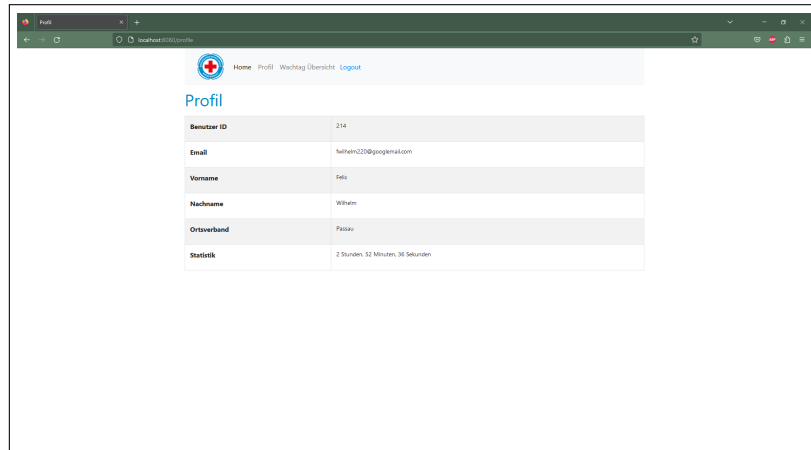
Abbildung 5.6.: Auszug aus header.html

### 5.2.4. Nutzerübersicht

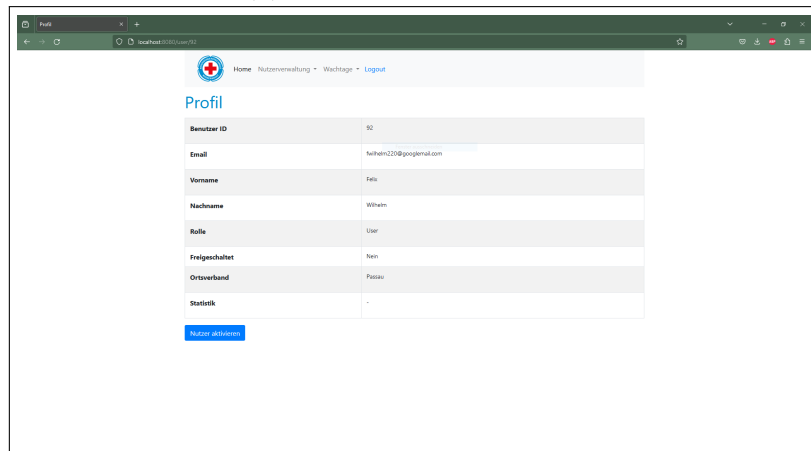
In der Anwendung haben Mitglieder die Möglichkeit ihre Profildaten einzusehen. Hier wird außerdem die Statistik ausgegeben, wie lange man schon im Wachdienst war. Die Statistik wird für jeden Nutzer berechnet, indem alle dem Nutzer zuordenbaren „UserGuardingRelationEntities“ gelesen werden. Innerhalb einer Entität kann dann mittels des Start- und Enddatums die jeweilige Dauer für den konkreten Wachtag berechnet werden.

Als Admin hat man zusätzlich eine Übersicht über alle registrierten Personen. Hier wird für jeden Nutzer die Statistik angegeben, die Rolle und ob der Nutzer schon aktiv ist. Wenn ein Nutzer noch nicht aktiviert wurde, kann ein Admin das Profil aufrufen und ihn über den Button „Freischalten“ bestätigen. So wird sichergestellt, dass keine unbefugten Personen Zugang zum System erhalten.

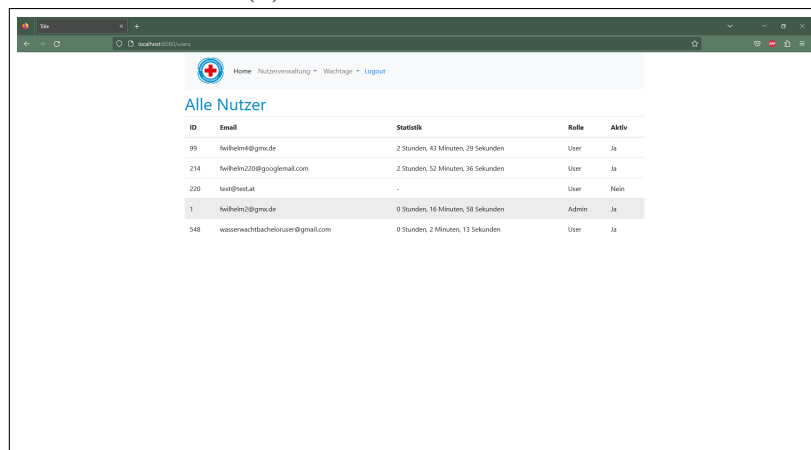
## 5. Konzeption und Umsetzung



(a) Profil - Nutzeransicht



(b) Profil - Adminansicht



(c) Nutzerübersicht

Abbildung 5.7.: Nutzerübersicht



### 5.2.5. Wachplanerstellung

Bei der Wachplanerstellung gibt es zum Einen die Möglichkeit für einen einzelnen Termin einen Wachtag zu erstellen. Hier muss ein Datum, sowie Start- und Endzeitpunkt angegeben werden. Aus diesen Daten wird eine „GuardDayEntity“ generiert und in der Datenbank gespeichert.

The screenshot shows a web browser window with the URL 'localhost:3000/guardday/create'. The page has a navigation bar with 'Home', 'Nutzerverwaltung', 'Wachtage', and 'Logout'. The main heading is 'Wachplan anlegen'. Below it are two tabs: 'Einzeltermin anlegen' (selected) and 'Terminreihe anlegen'. The form contains three input fields: 'Datum' with a placeholder 'TT.MM.JJJJ', 'Startzeit' with a placeholder 'HH:MM', and 'Endzeit' with a placeholder 'HH:MM'. At the bottom is a blue button labeled 'Wachtag speichern'.

Abbildung 5.8.: Einzeltermin anlegen

Es gibt außerdem die Möglichkeit gleich eine Reihe an Wachtagen generieren zu lassen. Dazu muss ein Start- und Enddatum der Reihe angegeben werden, sowie wieder ein Start- und Endzeitpunkt. Zusätzlich müssen die Tage ausgewählt werden, an denen ein Wachtag stattfinden soll. So wird für jeden ausgewählten Tag innerhalb diesen Zeitraums eine „GuardDayEntity“ abgeleitet und abgespeichert.

The screenshot shows the same web browser window, but the 'Terminreihe anlegen' tab is selected. The form contains four input fields: 'Start-Datum' (21.08.2023), 'End-Datum' (08.09.2023), 'Startzeit' (10:00), and 'Endzeit' (17:00). Below these fields is a row of checkboxes for the days of the week: Montag, Dienstag, Mittwoch, Donnerstag, Freitag, Samstag, and Sonntag. The 'Montag' through 'Freitag' checkboxes are checked. At the bottom is a blue button labeled 'Wachtag speichern'.

Abbildung 5.9.: Terminreihe anlegen

### 5.2.6. Wachplanübersicht

Auf der Wachtagübersicht (*siehe 5.10*) werden die verfügbaren Wachtage in Kalenderform dargestellt. Diese werden an ihrem Datum in ihren Start und Endzeitpunkte dargestellt. Zusätzlich gibt es für jeden Tag eine farbige Codierung um über den Zustand zu erfahren. Ist ein Wachtage rot hinterlegt, ist noch niemand für dieses Datum eingebucht. Ist ein Wachtage grün hinterlegt, ist mindestens eine Person eingebucht. Wenn ein Wachtage abgeschlossen wurde, wird er grau hinterlegt. Dies ermöglicht es den Leitungskräften auf einen Blick zu erkennen, bei welchen Wachtagen noch Personalmangel herrscht.

Um den Kalender darzustellen wurde das Open-Source Plugin „jquery-calendar“ der Firma „Arrobe“ verwendet. Dieser ließ sich sehr gut in die bestehende Anwendung integrieren und bietet die Möglichkeit, Termine farblich zu hinterlegen. Ein weiteres Feature dieses Kalenders ist das bestehende Responsive-Design. Der Kalender funktioniert auf einem Mobiltelefon ebenso gut wie auf einem Desktop-Computer. Über einen Klick auf ein Termin-Element gelangt man zur Seite des jeweiligen Wachtages.

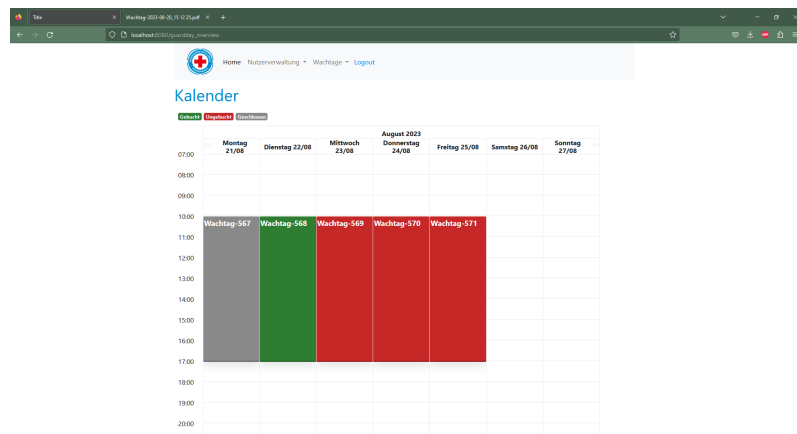


Abbildung 5.10.: Wachtagübersicht

### 5.2.7. Wachplandurchführung

Auf der Seite des jeweiligen Wachplans (*siehe 5.11*) befindet sich oben zunächst das geplante Datum und die geplanten Start- und Endzeitpunkte des Wachtages. Rechts daneben befindet sich die Anzeige der aktuellen Wetterdaten aus der Wetter-API, die über eine Grafik verdeutlicht, welche Wetterzustände gerade herrschen. Darunter befindet sich ein Eingabefeld der gemessenen Wassertemperatur. Die Eingabe ist jedoch erst möglich, wenn der Wachtage gestartet wurde.

Als nächstes findet man die Tabellen mit den gebuchten Helfern und den anwesenden Helfern. Diese können hinzugefügt werden, indem entweder aus der Dropdown-Liste „Nutzer wählen“ registrierte Nutzer ausgewählt werden, oder in dem Eingabefeld „Nutzer Eingabe“ als Helfer erfasst werden. Wenn ein Wachtage noch nicht begon-

## 5. Konzeption und Umsetzung

nen hat, wird ein Helfer als gebuchter Helfer gespeichert (*siehe 5.12*). Hier besteht noch die Möglichkeit, sich wieder aus einem Wachplan, über das Mülltonnen-Icon neben dem Namen, auszutragen. Sobald ein Wachtag begonnen hat, werden die so eingegebenen Nutzer als anwesende Helfer gespeichert.

Darunter befinden sich noch die Buttons „Wachbeginn“, um einen Wachtag zu starten, sowie der Button „Anmeldung ILS“, um eine An- und Abmeldung in der ILS zu dokumentieren. Der Button „Anmeldung ILS“ ist allerdings auch erst aktiv, wenn der Wachtag gestartet wurde.

Zuletzt findet man noch ein Textfeld, über das man Einträge in das Wachbuch schreiben kann. Hier besteht die Möglichkeit ein registriertes Mitglied aus der Liste „Nutzer wählen“ auszuwählen. Diese Person wird dann als Verfasser des Wachbucheintrags gespeichert. Wenn man keinen Nutzer aus der Liste auswählt, gilt der eingeloggte Anwender als Verfasser der Nachricht. In der darunterliegenden Tabelle sind die einzelnen Wachbucheinträge mit Zeitstempel und Nutzer chronologisch abgelegt.

The screenshot shows a web browser window with the URL `localhost:3000/wachtag_anmelden/547`. The page has a navigation bar with links: Home, Nutzerverwaltung, Wachtag, and Logout. The main heading is 'Wachplan'. Below it, the date '21.08.23' and time '10:00 - 17:00' are displayed. A weather icon shows a sun and clouds with a temperature of '28.3°C'. There is a 'Wassertemp.' input field and a 'Wasser speichern' button. The page is divided into two sections: 'Gebuchte Helfer' and 'Anwesende Helfer', both currently empty. Below these are two buttons: 'Wachtag beginnen' and 'Anmeldung ILS'. A 'Wachbuch Eintrag' section contains a large text area for notes, a 'Nutzer wählen' dropdown menu, and an 'Eintrag speichern' button. At the bottom, a table with columns 'Datum', 'Name', and 'Beschreibung' is shown, with the message 'Keine Einträge vorhanden' (No entries available).

Abbildung 5.11.: Wachtag Durchführung

## 5. Konzeption und Umsetzung

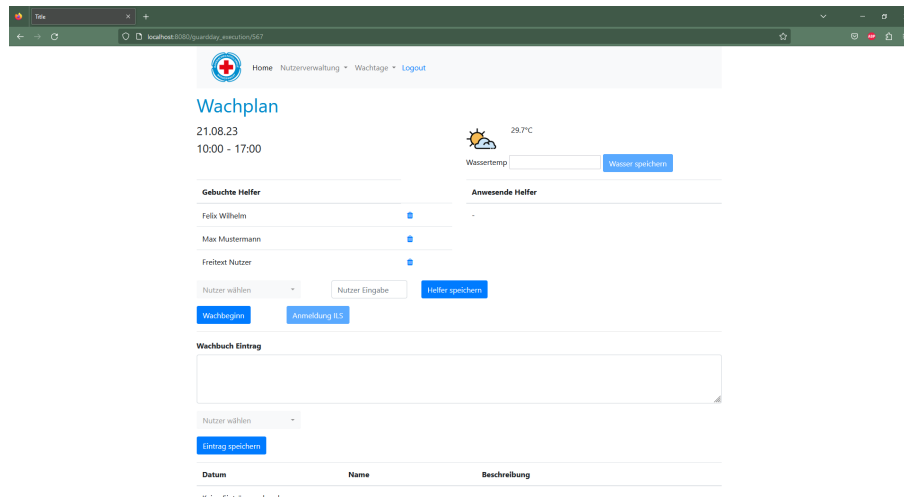


Abbildung 5.12.: Wachtag mit gebuchten Helfern

Sobald man einen Wachtag, über einen Klick auf „Wachbeginn“, startet, wird der erste Eintrag in das Wachbuch geschrieben um den Wachbeginn zu dokumentieren. In der Liste der gebuchten Helfer wird das Icon der Mülltonne entfernt, da man sich nach Wachstart nicht mehr aus dem Plan austragen kann. Stattdessen erscheint an der Stelle jetzt ein Pfeil-Icon. Wird dieses gedrückt, kann man den ausgewählten Nutzer in die Liste der anwesenden Helfer übertragen. Dies ermöglicht es den Mitgliedern, sich schnell für den Wachdienst anzumelden.

Wenn sich ein Nutzer nach Wachstart für den Wachdienst anmeldet, wird ein Eintrag in das Wachbuch geschrieben um den Wachbeginn zu dokumentieren. Ein anwesender Helfer kann sich vom Dienst abmelden, indem er das Mülltonnen Icon neben seinem Namen drückt. Dadurch wird auch wieder ein Eintrag generiert, um das Wachende festzuhalten (*siehe 5.13*).

Zusätzlich ist es jetzt möglich Wassertemperaturen, über das Eingabefeld oben, zu dokumentieren. Betätigt man den Button „Wasser speichern“, wird ein Eintrag in das Wachbuch geschrieben.

## 5. Konzeption und Umsetzung

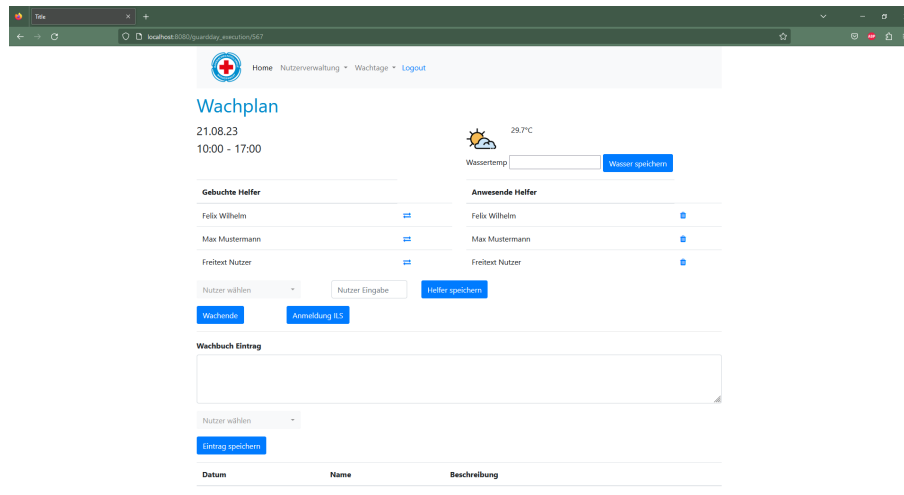


Abbildung 5.13.: Wachtag mit anwesenden Helfern

Die Wetterdaten werden automatisch in das Wachbuch geschrieben. Dies wurde umgesetzt, indem ein asynchroner Methodenaufruf abgesetzt wird. Dies bedeutet, dass die Methode aufgerufen wird, jedoch nicht auf das Ergebnis gewartet werden muss. In der Methode wird zunächst eine konfigurierte Zeit lang gewartet, bevor der Wachtag gelesen wird. Danach wird eine while-Schleife begonnen, die so lange läuft, bis der Wachtag beendet wurde. Innerhalb der Schleife werden dann die aktuellen Wetterdaten aus der Api gelesen und als Wachbucheintrag dem jeweiligen Wachtag beigelegt und gespeichert. Danach wird wieder eine konfigurierte Zeit gewartet, bevor der Wachtag erneut gelesen wird. Somit kann die Bedingung der while-Schleife erneut abgeprüft werden.

```
@Async
public void writeWeatherAPIResults(long guardDayId) throws InterruptedException {

    /*
    wait 1h
    read guard day
    while (guardDay is not closed)
        readWeatherApi
        writeWeatherApiResponse
        wait 1h
        read guardday;
    */
    Thread.sleep(120000);
    GuardDayDto guardDayDto = readGuardDayById(guardDayId);

    while (guardDayDto.getActualEndTime() == null) {
        WeatherApiResponse weatherApiResponse = weatherApi.getCurrentWeatherData();
        JournalEntryDto journalEntryDtoWeather = JournalHelper.createJournalEntry(guardDayDto.getGuardDayId(), EntryType.WEATHER, nu
guardDayDto.getJournalEntries().add(journalEntryDtoWeather);
        saveGuardDayDto(guardDayDto);
        Thread.sleep(10000);
        guardDayDto = readGuardDayById(guardDayId);
    }
}
```

Abbildung 5.14.: Auszug aus GuardDayService.java - Automatische Speicherung der Wetterdaten

Nach Beenden des Wachtages durch Klick auf „Wachende“ werden alle noch anwe-

## 5. Konzeption und Umsetzung

senden Helfer abgemeldet. Außerdem wird das Wachende im Wachbuch festgehalten. Alle Buttons für diesen Wachtage, bis auf das Wachbuch, werden deaktiviert, da ein beendeter Wachtage nicht mehr bearbeitet werden darf (*siehe* 5.15). Nach Wachende ist es immer noch möglich Einträge für das Wachbuch zu verfassen.

Einen beendeten Wachtage kann man sich im Nachhinein noch ansehen. So kann man über das Wachbuch den Verlauf des Wachtages nachvollziehen (*siehe* 5.16).

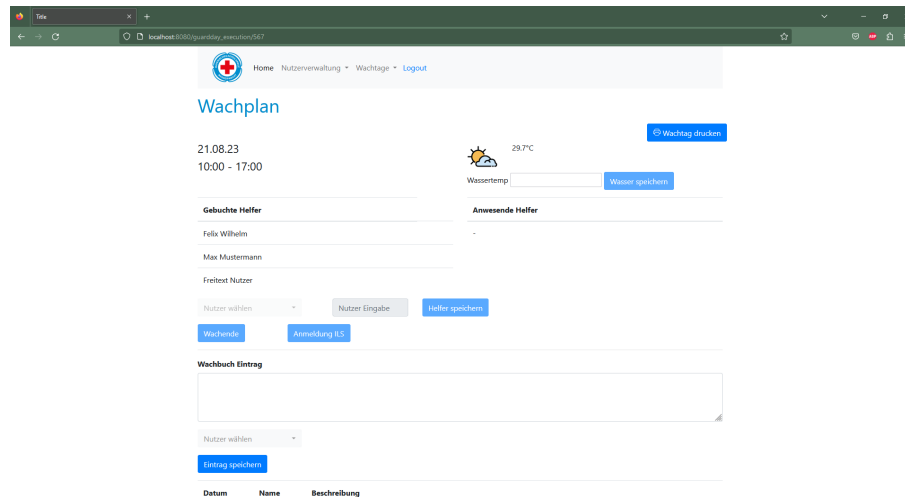
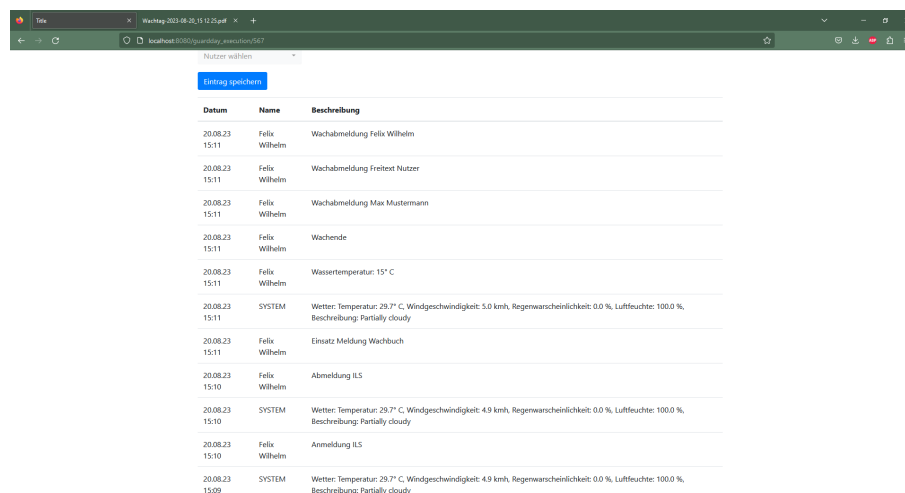


Abbildung 5.15.: Wachende



Datum	Name	Beschreibung
20.08.23 15:11	Felix Wilhelm	Wachabmeldung Felix Wilhelm
20.08.23 15:11	Felix Wilhelm	Wachabmeldung Freitext Nutzer
20.08.23 15:11	Felix Wilhelm	Wachabmeldung Max Mustermann
20.08.23 15:11	Felix Wilhelm	Wachende
20.08.23 15:11	Felix Wilhelm	Wassertemperatur: 15° C
20.08.23 15:11	SYSTEM	Wetter: Temperatur: 29.7° C, Windgeschwindigkeit: 5.0 km/h, Regenwahrscheinlichkeit: 0.0 %, Luftfeuchte: 100.0 %, Beschreibung: Partially cloudy
20.08.23 15:11	Felix Wilhelm	Einsatz Meldung Wachbuch
20.08.23 15:10	Felix Wilhelm	Abmeldung ILS
20.08.23 15:10	SYSTEM	Wetter: Temperatur: 29.7° C, Windgeschwindigkeit: 4.9 km/h, Regenwahrscheinlichkeit: 0.0 %, Luftfeuchte: 100.0 %, Beschreibung: Partially cloudy
20.08.23 15:10	Felix Wilhelm	Anmeldung ILS
20.08.23 15:09	SYSTEM	Wetter: Temperatur: 29.7° C, Windgeschwindigkeit: 4.9 km/h, Regenwahrscheinlichkeit: 0.0 %, Luftfeuchte: 100.0 %, Beschreibung: Partially cloudy

Abbildung 5.16.: Wachbuch

### 5.2.8. Druckfunktionalität

Nach Ende eines Wachtages besteht die Möglichkeit, sich ein PDF aus den Wachbucheinträgen zu generieren. Dazu wird ein Button oben rechts „Wachtage

## 5. Konzeption und Umsetzung

drucken“ (*siehe* 5.15) eingeblendet. Leitungskräfte haben so die Möglichkeit einen detaillierten Verlauf des Wachtages mit allen Geschehnissen abzulegen (*siehe* A.3). Umgesetzt wurde die Druckfunktionalität mittels der Open-Source Java-Library OpenPDF von LibrePDF. Diese ermöglicht es über Java Code eine PDF-Datei zu erzeugen.

## 6. Evaluation

Um sicherzustellen, dass die Website den Anforderungen der Wasserwacht entspricht, wird im Folgenden eine Evaluierung durchgeführt. Dies geschieht zunächst mit Nutzertest und Befragung von Wasserwachtmitgliedern. Dadurch soll die Praxistauglichkeit und Effektivität der Webanwendung bewertet werden. Außerdem werden noch „Klickmetriken“ betrachtet, für die zwei Haupt-Anwendungsfälle Wachplanerstellung und Wachplandurchführung. Abschließend wird daraus ein Fazit gezogen und auf Verbesserungspotentiale eingegangen.

### 6.1. Nutzertests

Für die Nutzertests wurden drei unabhängige Personen herangezogen. Diesen wurde zunächst der Prozess der Wachplanung der Wasserwacht erläutert und anhand der aktuellen Papierform (siehe Anhang [A.2](#)) aufgezeigt. Im nächsten Schritt wurde ihnen die Webanwendung präsentiert und die einzelnen Prozessschritte erläutert, die zur Wachplanung und Durchführung gehören.

Anschließend wurde ihnen ein Bewertungsbogen vorgelegt mit 11 Fragen, die auf die Benutzerfreundlichkeit und Praxistauglichkeit der Anwendung abzielen. Diese 11 Fragen waren mit Punkten von 1 bis 5 zu bewerten, wobei 1 sehr schlecht entspricht und 5 sehr gut. Zusätzlich gab es noch zwei offene Fragen, um Verbesserungspotentiale und Kritik abzufragen.

Aus den Bewertungsfragen wird anschließend ein Score berechnet, der den Durchschnitt über alle Fragen abbildet. Ziel ist es hier über den Mittelwert von 33 zu kommen. Dieser würde sich ergeben wenn alle Fragen mit 3 bewertet würden.

Wird dieses Ziel erreicht, wird von einer benutzerfreundlichen Website ausgegangen.

#### 6.1.1. Fragenkatalog

##### Bewertungsfragen

Zunächst wurden Fragen definiert, die auf die generelle Nutzerzufriedenheit mit der Website abzielen. Dazu wird der Nutzer zu Design, Navigation, Verständlichkeit und Zugänglichkeit von Informationen befragt.

1. Wie zufrieden waren Sie mit dem Design und Layout der Website?
2. Wie leicht war es für Sie, die Informationen zu finden, nach denen Sie gesucht haben?
3. Wie einfach war es für Sie, sich auf der Website zu orientieren?
4. Wie benutzerfreundlich fanden Sie die Navigation auf der Website?
5. Wie gut funktionieren die interaktiven Funktionen auf der Website (z.B. Suchfeld,



Formulare, ...) ?

6. Wie verständlich sind die Hinweise und Anweisungen auf der Website?

Im nächsten Teil, zielen die Fragen auf die Zufriedenheit mit der Umsetzung der Prozesse ab. Dadurch soll man erkennen, ob durch die digitalen Prozesse ein Mehrwert entstanden ist im Vergleich zum analogen Vorgehen.

7. Wie einfach ist es einen Wachplan anzulegen?

8. Wie einfach ist es einen Wachplan durchzuführen?

9. Wie schnell war der digitale Verwaltungsprozess im Vergleich zum analogen Verfahren?

10. Wie zufrieden sind Sie mit dem Ergebnis des digitalen Verwaltungsprozesses im Vergleich zum analogen Verfahren?

11. Wie wahrscheinlich ist es, dass Sie die digitale Version des Verwaltungsprozesses im Vergleich zur analogen Version nutzen möchten?

### Offene Fragen

1. Wie könnte man die Website noch verbessern?

2. Gibt es noch abschließendes Feedback / Kritik?

### 6.1.2. Ergebnisse

In Tabelle 6.1 sind die Ergebnisse der einzelnen Fragen aufgezeigt. Als Ergebnis wird immer der durchschnittliche Score aus den 3 Umfrageteilnehmern genommen.

Addiert man diese Ergebniswerte auf, erhält man einen Score von 49,34. Somit wurde das Ziel einer benutzerfreundlichen Webanwendung erreicht.

Tabelle 6.1.: Ergebnisse der Nutzerumfrage

Frage	Ergebnis (Durchschnitt)
1	4
2	4,67
3	4,67
4	4,33
5	4,67
6	4,33
7	5
8	3,67
9	5
10	4,67
11	4,33

Für die offenen Fragen gab es folgende Antworten:

### **Wie könnte man die Website noch verbessern?**

1. Die Farbgebung ist etwas eintönig, was allerdings Sinn macht, da es den Farben des Wasserwacht Logos folgt.
2. Design eher monoton, Buttons heben sich nicht gut hervor, Abgrenzung schwierig.

### **Gibt es noch abschließendes Feedback / Kritik?**

1. Sehr übersichtlich und einfach gehalten, dass einzige was fehlt ist die Anzahl der Schwimmer, Boote etc. am aktuellen Tag in der Übersicht.
2. Die Website ist äußerst übersichtlich, mit Details an der richtigen Stelle. Die Durchführung verschiedener Aktionen ist logisch abgebildet, was die Benutzung angenehm und einfach macht. Äußerst hervorzuheben ist die Zwei-Phasen Authentifizierung zum Schutz der Daten.
3. Druckfunktionalität ist gut, alle Informationen können per PDF archiviert oder versendet werden. Sehr übersichtlich.

### **6.1.3. Diskussion**

Grundsätzlich kann man sehen, dass in der Umfrage positive Ergebnisse bzgl der Website erfasst wurden. Sowohl im ersten Fragenteil zur Benutzerfreundlichkeit, als auch im zweiten Fragenteil zur Prozess-Zufriedenheit konnten hohe Werte erzielt werden.

Dadurch lässt sich feststellen, dass auch Personen ohne fachliche Expertise sich einfach auf der Website zurecht finden. Die Website ist sehr schlicht aufgebaut, wodurch die Personen die Prozesse der Wachplanung und Wachdurchführung sehr schnell verstanden haben. Dies spiegelt sich auch in den offenen Fragen wieder, in denen die Übersichtlichkeit gelobt wird.

Ein weiterer Punkt aus den offenen Fragen bemängelt das Fehlen eines digitalen Einsatztagebuchs. Dieses kann man am aktuellen Wachbuch in Papierform sehen und beinhaltet verschiedene Einsatzdaten, wie zum Beispiel wie viele Einsätze mit Boot abgeschlossen wurden.

Das Einsatztagebuch ist auch im initialen Entwurf der Plansoftware (s.h. [A.1](#)) zu sehen. Aus zeitlichen Gründen wurde sich in dieser Bachelorarbeit nur mit der Wachplanung und Durchführung befasst. Als Erweiterungsmöglichkeit wäre es aber durchaus wichtig.

Weiterhin wird das monotone Design kritisiert bezüglich eintöniger Farbgebung und fehlender Hervorhebung von Formularen.

Für die Bachelorarbeit wurde der Fokus eher auf eine korrekte Umsetzung der Prozesse gelegt. Aus den Umfrageergebnissen zeigt sich aber, dass das Design auch eine wichtige Rolle spielt. Dies kann daher als Verbesserungsmöglichkeit in zukünftiger Weiterentwicklung mit aufgenommen werden.

Obwohl für die Nutzertests eine sehr kleine Personengruppe herangezogen wurde, waren sie dennoch sehr aufschlussreich. Besonders über das offene Feedback lie-

ßen sich die unterschiedlichen Vorstellungen verschiedener Personen einfangen. Für einen weiteren Durchlauf wäre eine größere Personengruppe von Vorteil, um weitere Einblicke in die Präferenzen der Nutzer zu erfassen.

## 6.2. Klickmetriken

Im Folgenden werden Klickmetriken gemessen und analysiert für die zwei Hauptprozesse Wachplanerstellung und Wachplandurchführung. Dazu wird betrachtet, wie viele Klicks man benötigt um zur gewünschten Zielseite zu gelangen. Weiterhin werden dann die Klicks gemessen die für die Durchführung der Prozessschritte benötigt werden.

### 6.2.1. Wachplanerstellung

Für die Wachplanerstellung wird als Ausgangspunkt die Startseite festgelegt.

Um zur Seite der Wachplanerstellung zu gelangen werden zwei Klicks benötigt. Zunächst klickt man in der Navigationsleiste auf „Wachtag Übersicht“ und dann auf den Unterpunkt „Wachtagerstellung“.

Für die Anlage eines einzelnen Wachtages werden 4 Klicks benötigt. Diese ergeben sich aus den jeweiligen Eingabefeldern „Datum“, „Startzeit“ und „Endzeit“ und einem Klick zum speichern.

Für die Anlage einer Terminreihe werden 7 - 13 Klicks benötigt. Zunächst muss man auf „Terminreihe anlegen“ klicken um zur Anlage zu gelangen. Danach werden die Eingabefelder „Startdatum“, „Enddatum“, „Startzeit“ und „Endzeit“ befüllt. Die Auswahl der Tage, an denen ein Wachtag erzeugt werden soll, wurde mit Hilfe von Checkboxen umgesetzt. Daher sind hier 1 - 7 Klicks notwendig. Abschließend muss man die Reihe noch speichern.

### 6.2.2. Wachplandurchführung

Für die Durchführung eines Wachtags wird ein einfacher Nutzer verwendet. Als Ausgangspunkt gilt wieder die Startseite.

Um zu dem jeweiligen Wachtag zu gelangen werden drei Klicks benötigt. Zunächst klickt man in der Navigationsleiste auf „Wachtagübersicht“. Danach wählt man einen Wachtag im Kalender aus und bestätigt die Auswahl in einem sich öffnendem Modal.

Da die Wachplandurchführung je nach Wachtag unterschiedlich ist und auch nicht in einer Sitzung alle Punkte abgearbeitet werden, wird im Folgenden auf die Klicks der einzelnen Prozessschritte eingegangen.

### **Einbuchung**

Um sich für einen noch nicht gestarteten Wachtag einzubuchen klickt man auf die Liste „Nutzer wählen“ und wählt den entsprechenden Nutzer aus. Durch Klick auf „Helfer speichern“ wird der Nutzer für den Wachtag hinterlegt. Somit ergeben sich drei Klicks.

### **Wachbeginn**

Für den Wachbeginn werden zwei Klicks benötigt. Zunächst startet man den Wachtag durch Klick auf „Endzeit“ Wachbeginn. Danach kann man in der Liste der gebuchten Helfer durch Klick auf das Pfeil-Icon einen Nutzer anmelden.

### **Wassertemperatur messen**

Um die Wassertemperatur zu dokumentieren werden zwei Klicks benötigt. Zunächst klickt man in das Eingabefeld der Wassertemperatur. Anschließend wird der Eintrag gespeichert.

### **An- / Abmeldung ILS**

Für die An- und Abmeldung im ILS werden zwei Klicks benötigt.

### **Wachbuch Eintrag**

Um einem Wachbuch Eintrag anzulegen klickt man zunächst in das Eingabefeld „Wachbuch Eintrag“. Nach erfassen des Eintrags speichert man das ganze durch Klick auf „Eintrag speichern“. Das entspricht zwei Klicks.

### **6.2.3. Interpretation**

Insgesamt kann festgestellt werden dass Nutzer der Website nur geringe Anzahlen an Klicks aufwenden müssen, um ihre gewünschten Ziele zu erreichen. Dies gibt einen Einblick in die minimalistische Gestaltung der Website und ihr schlankes Design.

Durch das minimalistische Design werden überflüssige und ablenkende Elemente eliminiert, wodurch eine klare und fokussierte Benutzererfahrung entsteht. Dies führt zu einer effizienteren Navigation und Interaktion, da Nutzer weniger Schritte durchlaufen müssen, um Prozesse abzuarbeiten.

Die Website ist so gestaltet, dass sie intuitiv und leicht zu bedienen ist. Indem die Anzahl der erforderlichen Klicks minimiert wird, wird der Zugang zu Informationen und Funktionen erleichtert, was besonders für Nutzer mit geringerer technischer Erfahrung von Vorteil ist.

## 6.3. Verbesserungspotentiale

### **Fehlende Bearbeitungsmöglichkeiten in Profil und Wachplanung**

Derzeit können Benutzer einen einmal erstellten Wachtag nicht löschen oder bearbeiten. Zudem fehlt auch die Bearbeitungsmöglichkeit im Nutzerprofil.

Hierzu müssen noch die fehlenden Funktionen implementiert werden und durch zusätzliche Schaltflächen oder interaktive Elemente in der Nutzeroberfläche eingebunden werden. Eine solche Erweiterung würde die Benutzerfreundlichkeit erhöhen und den Benutzern mehr Flexibilität bieten.

### **Monotones Design**

Das aktuelle Design der Webanwendung nutzt Standard Bootstrap, was zu einem eher monotonen Erscheinungsbild führt.

Hier könnte man das Design weiter personalisieren durch Anpassung von Bootstrap-Themen oder Integration von CSS-Stilen, die über die Standardvorlagen hinausgehen. Durch eigene Farbschemata, Schriftarten und Layout Änderungen könnte man das Design so anpassen, dass es genau auf die Wasserwacht zugeschnitten wird.

### **Inkonsistentes responsives Design in der Wachplandurchführung**

Die Webanwendung ist auf mobilen Geräten nur eingeschränkt nutzbar, da viele Felder auf der Seite der Wachplandurchführung vorhanden sind. Die Darstellung auf kleineren Bildschirmen wird dabei unübersichtlich.

Durch den Einsatz von MediaQueries könnte man hier das Design noch weiter optimieren, um auch den Gebrauch auf mobilen Geräten zu gewährleisten.

## 6.4. Ausblick auf zukünftige Entwicklungen und mögliche Erweiterungen der Webanwendung

Die Webanwendung für die Wasserwacht Dingolfing-Landau hat bereits einen bedeutenden Beitrag zur Digitalisierung und Effizienzsteigerung der Wachplanung und -durchführung geleistet. Dennoch bieten sich verschiedene Möglichkeiten zur Weiterentwicklung und Erweiterung, um den sich ändernden Anforderungen und Möglichkeiten gerecht zu werden.

### **Integration fehlender Elemente aus dem Planungsmodell**

Aus zeitlichen Gründen wurden manche Punkte aus dem ursprünglichen Entwurf der Anwendung nicht umgesetzt. Die Integration einer detaillierten Übersicht über

verschiedene Checks sowie eines digitalen Einsatztagebuchs würde jedoch die Funktionalität der Anwendung erweitern. Dies könnte die Nachverfolgung und Dokumentation von Ausrüstungschecks und spezifischen Ereignissen während der Wache umfassen, was eine wertvolle Ressource für die Analyse und das Reporting darstellt.

### **Erweiterung der Nutzerprofile**

Eine wichtige Erweiterung wäre die Möglichkeit für Nutzer, ihre Qualifikationen und Zertifikate in ihren Profilen zu pflegen. Dies würde eine effizientere Zuweisung von Aufgaben ermöglichen, da die Verantwortlichen schnell die geeignetsten Mitglieder für spezifische Einsätze identifizieren können.

### **Integration mit weiteren Systemen**

Die Interoperabilität mit anderen digitalen Systemen der Wasserwacht könnte die Effizienz weiter steigern. Beispielsweise könnte eine Verbindung mit internen Kommunikationssystemen oder Datenbanken die Koordination und Informationsverteilung verbessern.

### **Datenschutz und Sicherheit**

Der Datenschutz ist ein kritischer Aspekt, der in der ursprünglichen Implementierung nicht ausreichend berücksichtigt wurde. Es ist notwendig, die Anwendung in Übereinstimmung mit den Datenschutzgesetzen, wie der DSGVO, zu überarbeiten. Dies beinhaltet die Implementierung von Funktionen zur Datensicherheit und die Gewährleistung der Transparenz in der Datenverarbeitung.

### **Technische Infrastruktur und Sicherheit**

Für eine verbesserte Performance und Zuverlässigkeit sollte die Anwendung auf einem dedizierten Server installiert werden. Zusätzlich ist ein robustes System zur Datensicherung und -wiederherstellung von entscheidender Bedeutung, um Datenverlust zu vermeiden und die Kontinuität des Betriebs zu gewährleisten.

### **Fazit**

Diese potenziellen Erweiterungen und Entwicklungen würden nicht nur die Funktionalität der Webanwendung erhöhen, sondern auch die allgemeine Effizienz und Effektivität der Operations- und Planungsprozesse der Wasserwacht Dingolfing-Landau verbessern. Durch kontinuierliche Anpassungen und Verbesserungen kann die Anwendung den sich wandelnden Bedürfnissen der Nutzer gerecht werden und einen wesentlichen Beitrag zur digitalen Transformation der Wasserwacht leisten.

## **7. Fazit**

### **7.1. Zusammenfassung der Arbeit**

Ziel dieser Arbeit war es, die Herausforderungen der Wachplanerstellung der Wasserwacht zu identifizieren und durch den Einsatz von IT-Lösungen zu bewältigen. In Kapitel 2 wurden die theoretischen Hintergründe dieser Arbeit aufgegriffen. Es wurde zum einen die Wasserwacht allgemein beschrieben, sowie auf Herausforderungen bei der Wachplanerstellung eingegangen.

In Kapitel 3 wurden die genauen Anforderungen an die Webanwendung festgemacht. Dazu wurden in mehreren Treffen mit Wasserwachtmitgliedern die Nutzergruppen festgelegt und die Prozesse der Wachplanung und Wachbuchführung festgehalten. Mit Hilfe von Mockups wurden die Ergebnisse daraus veranschaulicht. So hatte man bereits im frühen Entwicklungsstadium einen Überblick, wie die Webanwendung aussehen wird.

Nachfolgend wurden die geeigneten Technologien zur Umsetzung der Website ausgewählt. Diese wurden anhand von persönlicher Erfahrung, sowie Eignung für das Projekt ermittelt. In Kapitel 4 werden diese näher beschrieben.

In Kapitel 5 wird dann näher auf die Implementierung eingegangen. Zunächst wird das zu Grunde liegende Datenmodell anhand eines UML Diagramms beschrieben. Daraufhin werden die umgesetzten Funktionalitäten und Prozesse dargestellt.

Abschließend wurde noch eine Evaluation durchgeführt. Zum einen wurde ein Nutzertest durchgeführt, wobei einer Personengruppe die Webanwendung vorgestellt wurde und im Anschluss ein Bewertungsbogen ausgefüllt wurde. Weiterhin wurden auch noch Klickmetriken innerhalb der Website erfasst und analysiert.

Daraus lies sich ein gutes Feedback für die Anwendung ziehen sowie Problemstellen identifizieren.

### **7.2. Schlussfolgerungen und Handlungsempfehlungen**

Mit dieser Bachelorarbeit wurde aufgezeigt, dass man analoge Verwaltungsprozesse sehrwohl als digitale Lösungen abbilden kann. Dazu bedarf es einer intensiven Planung mit fachlich vertrauten Personen.

Um sicherzustellen, dass diese Prozesse zur Zufriedenheit der Nutzer korrekt umgesetzt werden, braucht es eine Testphase. Im Kontext der Webanwendung für die Wasserwacht, wäre die Empfehlung das System auf einem den Mitgliedern zugänglichem Server zu installieren. Über eine längere Testphase könnte man so mit dem System arbeiten und vertraut werden. Dadurch könnte man Erfahrungen sammeln und auch mögliche Schwachstellen weiter identifizieren.

Weiterhin wäre es wichtig auf eine Datensicherung zu achten. Um sicher zu gehen, dass die erfassten Daten der Wachtage nicht verloren gehen, könnte man re-


## 7. *Fazit*

gelmäßig Datenabzüge machen und entweder über verschiedene RAID-Systeme oder über Cloudanbieter sicher speichern.  
So kann sichergestellt werden, dass die Digitalisierung von Verwaltungsprozessen optimal verläuft.



## A. Anhang


### A.1. Initialer Entwurf

 <p><b>Kreiswasserwacht Dingolfing-Landau</b> Krankenhausstraße 6 • 84130 Dingolfing</p>		Projekt: <b>Verwaltungs-Software</b>		Erstellt am: 12.01.2022	
		Zeichnung: <b>Modul Wachbuch - Seite 1</b>		Erstellt von: Andreas Schmeisl	
				Seite: Verzeichnis: unwww	
Revision 1:	Änderung:	Revision 2:	Änderung:	Revision 3:	Änderung:

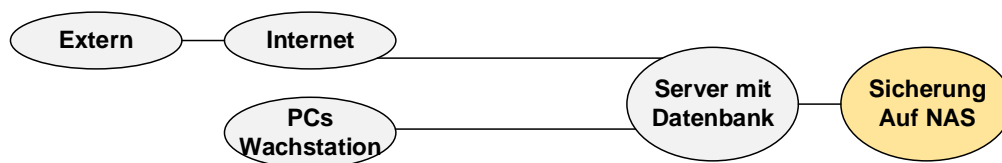
  

### Funktionen

Eingabemaske / Frontend Mitglieder / Helfer		Auswertungen / Backend Leitungskräfte / Admins		App/Onlinezugang Mitarbeiter / Helfer	
<b>Zeiterfassung</b>					
Chip		Anlegen Commissionen		Terminbuchung	
Eingabe über Frontend		Zuordnung Commissionen			
		Auswertung			
		Helferstunden			
<b>Wachplanung</b>					
Terminbuchung				Terminbuchung	
		Terminplanung			
<b>Materialcheck</b>					
Durchführung				Durchführung	
		Anlegen von Checklisten			
		Archivierung			
Bestellung / QM					
		Statistiken Verbrauch etc.			
<b>Wachbuch</b>					
Wachtag öffnen/schließen		Auswertung			
Wetterdaten ggf. von KNX		Wachtage			
Übersicht gebuchte Helfer		Wachstunden			
Übersicht eingestempelt					
Einsatztagebuch					
Übersicht Checks durchgef.					
<b>Sanitätsbuch</b>					
Einsätze eingeben		Einsätze verwalten		Einsätze eingeben	
Protokoll drucken		Archivieren		Protokoll drucken	
		Statistik			

 <p><b>Kreiswasserwacht Dingolfing-Landau</b> Krankenhausstraße 6 - 84130 Dingolfing</p>	Projekt: <b>Verwaltungs-Software</b>			Erstellt am: 12.01.2022	
	Zeichnung: <b>Modul Wachbuch - Seite 2</b>			Erstellt von: Andreas Schmeisl	
				Seite:	
				Verzeichnis: u:\www	
Revision 1:	Änderung:	Revision 2:	Änderung:	Revision 3:	Änderung:


## Struktur



## Benutzer

Admin
Funktionen
Qualifikationen, gültig bis...
Mail
Handy

Helfer
Funktionen
Qualifikationen, gültig bis...
Mail
Handy

 <p><b>Kreiswasserwacht Dingolfing-Landau</b> <small>Krankenhausstraße 6 - 84130 Dingolfing</small></p>	Projekt: <b>Verwaltungs-Software</b>				Erstellt am: 12.01.2022	
	Zeichnung: <b>Modul Wachbuch - Seite 3</b>				Erstellt von: Andreas Schmeisl	
					Seite:	
					Verzeichnis: u:www	
Revision 1:	Änderung:	Revision 2:	Änderung:	Revision 3:	Änderung:	

## Muster Checkliste

Fahrzeug / Raum / ...							
Datum/Uhrzeit							
	soll	ist					
Fach 1							
Wurfsäcke	2	2	i.O.	fehlt	Erl.		Bemerkung
Flossen	2	2	i.O.	fehlt	Erl.		Bemerkung
Schnorchel	2	2	i.O.	fehlt	Erl.		Bemerkung
Füßlinge	2	2	i.O.	fehlt	Erl.		Bemerkung

Fehlt: Automatische Weiterleitung an TL  
Erl.: wenn sofort aufgefüllt oder bei Nachbearbeitung (nur noch Admins)

Speichern → Unterschrift

Wird gespeichert und mit grünem Punkt in Übersicht markiert, wenn alles i.o.  
 Ansonsten zu offenen Aufträgen für TL's  
 Check wird auch im Wachbuch erfasst und als erledigt markiert

## Muster Wachbuch

Datum, Uhrzeit

Wetter (aus KNX) aktuell

Verlauf

Drucken

Wassertemp. Eingabe

speichern

Gebuchte Helfer (lt. Plan)

Anwesende Helfer  
(lt. Stempelung)

Zum Wachplan

Name

Ein/ausbuchen

An/-Abmeldung ILS

Wachbeginn/-ende

Wachbeginn/-ende

Übersicht Checks

Einsätze (sep. Wachbuch)

Name

Ereignis eintragen

speichern

Dat/Zeit	Name	Ereignis
Dat/Zeit	Name	Ereignis
Dat/Zeit	Name	Ereignis

Abbildung A.1.: Initialer Entwurf

## A.2. Wachbuch Papier



Kreiswasserwacht  
Dingolfing-Landau

Wachstation Mamming  
Wachbuch 2022

04 Wachstation Mamming  
04.001



Blatt	Datum	Beginn	Ende	Dauer	Anmeldung	Abmeldung
22 / 01	. .2022	:	:	:	:	:

Witterung	Uhrzeit	Wetter	Wasser	Luft
Vormittag	:	<input type="checkbox"/> sonnig <input type="checkbox"/> bewölkt <input type="checkbox"/> Regen <input type="checkbox"/> Sturm	°C	°C
Mittag	:	<input type="checkbox"/> sonnig <input type="checkbox"/> bewölkt <input type="checkbox"/> Regen <input type="checkbox"/> Sturm	°C	°C
Nachmittag	:	<input type="checkbox"/> sonnig <input type="checkbox"/> bewölkt <input type="checkbox"/> Regen <input type="checkbox"/> Sturm	°C	°C

Besprechung	Stations-Check	Fahrzeug-Check
:	<input type="checkbox"/> i.O.	<input type="checkbox"/> i.O.

Leistungsbericht	Tag	Übertrag Vorblatt	Gesamt	Bemerkungen
EH-Leistungen (Land/San)				
WR-Einsätze				
davon Schwimmer				
Davon Surfer / SUP				
Davon Boote (Schlauchboote etc.)				
mit Boot				
mit Fahrzeug/Quad				
mit Tauch-Einsatz				
Naturschutz- Beanstandungen				
Wachstunden *	:	:	:	

Wachleiter	Von	Bis	Dauer	Bemerkungen
1	:	:	:	
2	:	:	:	

Wachpersonal/Helfer	Von	Bis	Dauer	Bemerkungen
3	:	:	:	
4	:	:	:	
5	:	:	:	
6	:	:	:	
7	:	:	:	
8	:	:	:	
9	:	:	:	
10	:	:	:	
11	:	:	:	
12	:	:	:	
<b>Gesamt *</b>			:	

## A. Anhang

[illegible]

Kreiswasserwerk Dingolfing-Landau	Version:	1.0	Erstellt:	A. Schmeisl (B.Eng.), 05.05.2022	EQ-Freigabe:	A. Schmeisl (B.Eng.), 05.02.2022
04.001 Wachbuch 2022	Ersetzt:	-	Fachl. Freigabe:	A. Schmeisl (B.Eng.), 05.05.2022	Seite:	2 von 2

Abbildung A.2.: Wachbuch Papier

**Wachtag Export - 2024-01-19**

Datum	Beginn	Ende	Dauer
2024-01-19	21:25	21:27	0H 2M 10S

**Wetterdaten**

Uhrzeit	Typ	Eintrag
21:26	Wetter	Wetter: Temperatur: -0.4° C, Windgeschwindigkeit: 8.0 kmh, Regenwahrscheinlichkeit: 0.0 %, Luftfeuchte: 93.6 %, Beschreibung: Overcast
21:26	Wassertemperatur	Wassertemperatur: 50° C
21:27	Wetter	Wetter: Temperatur: -0.4° C, Windgeschwindigkeit: 8.2 kmh, Regenwahrscheinlichkeit: 0.0 %, Luftfeuchte: 93.4 %, Beschreibung: Partially cloudy

**Wachpersonal**

Name	Von	Bis	Dauer
User User	21:25	21:26	0H 1M 27S
Felix Wilhelm	21:25	21:27	0H 1M 27S

**Wachbuch**

Typ	Uhrzeit	Melder	Ereignis
Wachbeginn	21:25	Admin Admin	Wachstart
Meldung	21:25	Admin Admin	fdxgxvb
Anmeldung ILS	21:26	Admin Admin	Anmeldung ILS
Abmeldung ILS	21:26	Admin Admin	Abmeldung ILS
Wachende	21:27	Admin Admin	Wachende

Abbildung A.3.: Wachtag Export

## Literaturverzeichnis

- [AM17] Fernando Almeida and José Monteiro. The role of responsive design in web development. *Webology*, 14(2), 2017.
- [booa] Bootstrap - grid system. <https://getbootstrap.com/docs/5.3/layout/grid/> (besucht am 05.01.2024).
- [boob] Bootstrap - javascript. <https://getbootstrap.com/docs/4.6/getting-started/javascript/> (besucht am 05.01.2024).
- [brk] Brk wasserwacht. <https://www.brk.de/wasserwacht.html> (besucht am 05.01.2024).
- [Cho] Choi. Responsive design provides the best user experience. <https://www.activewebgroup.com/responsive-design-provides-best-user-experience/> (besucht am 05.01.2024).
- [drka] Drk wasserwacht - angebote. <https://www.drk.de/wasserwacht/angebote/> (besucht am 05.01.2024).
- [drkb] Drk wasserwacht - Über uns. <https://www.drk.de/wasserwacht/ueber-uns/> (besucht am 05.01.2024).
- [fig] Figma - prototyping. <https://www.figma.com/de/prototyping/> (besucht am 05.01.2024).
- [Kad12] Tim Kadlec. *Implementing Responsive Design: Building sites for an anywhere, everywhere web*. New Riders, 2012.
- [Nie] Jakob Nielsen. Usability 101: Introduction to usability. <https://www.nngroup.com/articles/usability-101-introduction-to-usability/> (besucht am 05.01.2024).
- [Ott] Mark Otto. Building twitter bootstrap. <https://alistapart.com/article/building-twitter-bootstrap/> (besucht am 05.01.2024).
- [pos] Postgres - about. <https://www.postgresql.org/about/> (besucht am 05.01.2024).
- [sca] When and why use java in web development? <https://www.scalosoft.com/blog/when-and-why-use-java-in-web-development/> (besucht am 05.01.2024).
- [spra] Introduction to the spring ioc container and beans. <https://docs.spring.io/spring-framework/reference/core/beans/introduction.html> (besucht am 15.08.2023).

## LITERATURVERZEICHNIS

- [sprb] Spring boot overview. <https://spring.io/projects/spring-boot#overview> (besucht am 15.08.2023).
- [sprc] Spring dta jpa. <https://spring.io/projects/spring-data-jpa/> (besucht am 05.01.2024).
- [sprd] Spring dta jpa - persisting entities. <https://docs.spring.io/spring-data/jpa/reference/jpa/entity-persistence.html> (besucht am 05.01.2024).
- [Spu13] Jake Spurlock. *Bootstrap: responsive web development*. O'Reilly Media, Inc., 2013.
- [thya] Thymeleaf - tutorial: Thymeleaf + spring. <https://www.thymeleaf.org/doc/tutorials/3.1/thymeleafspring.html> (besucht am 15.08.2023).
- [thyb] Thymeleaf - tutorial: Using thymeleaf. <https://www.thymeleaf.org/doc/tutorials/3.1/usingthymeleaf.html> (besucht am 15.08.2023).
- [wea] Visual crossing - weather api. <https://www.visualcrossing.com/weather-api/> (besucht am 05.01.2024).



Wilhelm, Felix

Passau, den 18.01.2024

Lindental 9

94032, Passau

Tel. Nr.: 0160/7569206

## ERKLÄRUNG

Ich erkläre, dass ich die vorliegende Arbeit mit dem Titel „Prozessdesign- und Digitalisierung für eine moderne Wasserwachtverwaltung“ selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel verfasst habe und dass alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, als solche gekennzeichnet sind.

Mit der aktuell geltenden Fassung der Satzung der Universität Passau zur Sicherung guter wissenschaftlicher Praxis und für den Umgang mit wissenschaftlichem Fehlverhalten vom 31. Juli 2008 (VABIUP Seite 283) bin ich vertraut.

Ich erkläre mich einverstanden mit einer Überprüfung der Arbeit unter Zuhilfenahme von Dienstleistungen Dritter (z.B. Anti-Plagiatssoftware) zur Gewährleistung der einwandfreien Kennzeichnung übernommener Ausführungen ohne Verletzung geistigen Eigentums an einem von anderen geschaffenen urheberrechtlich geschützten Werk oder von anderen stammenden wesentlichen wissenschaftlichen Erkenntnissen, Hypothesen, Lehren oder Forschungsansätzen.

---

Name, Vorname