

Qualitätssicherung

Intuitive Time Tracking

Betreuer
Maximilian Li

Team
Dominik Pollok
Phil Gengenbach
Alina Petri
José Ayala
Johann Kohl

Inhaltsverzeichnis

1	Einleitung	1
2	Bugfixing	1
2.1	Frontend	1
2.1.1	Styling	1
2.1.2	Popups	2
2.1.3	Timesheet	3
2.1.4	Users	4
2.1.5	Pages	5
2.1.6	Input and Error Handling	6
2.1.7	Further Issues	7
2.2	Backend	7
2.2.1	Document Generation	7
2.2.2	Timesheet	8
2.2.3	Users	11
2.2.4	Time Entry	13
2.2.5	DB Setup	15
2.2.6	Further Issues	15
3	Frontend Testing Routine	16
3.1	Initialisierung und Setup-Prüfung	16
3.2	Benutzerverwaltung: FrontendSupervisor	16
3.3	Benutzerverwaltung: FrontendHiwi	17
3.4	Archivierung und Reaktivierung von Benutzern	17
3.5	Funktionsprüfung: FrontendHiwi	18
3.6	Funktionsprüfung: FrontendSupervisor	19
3.7	Funktionsprüfung: FrontendSecretary	20
3.8	Überstundenberechnung und Übertrag	20
4	New Features	20
4.1	User-Archivierung	20
4.2	Einführung von Activity Type	21
4.3	Notification Feature	21
4.3.1	In-App Benachrichtigungen	21
4.3.2	Slack Integration	22
4.4	Dynamic Profile Picture Hashing	23
4.5	Fuzzy Search	23

5	Deployment	24
5.1	Dockerisierung der Anwendung	24
5.2	Konfiguration und Kommunikation der Container	25
5.3	Bereitstellung auf dem Zielserver	25
6	Testing	25

1 Einleitung

In diesem Dokument werden die im Rahmen der Qualitätssicherung durchgeführten Maßnahmen detailliert beschrieben. Unser primäres Ziel war es, eine Anwendung zu entwickeln, die durch hohe Korrektheit, Zuverlässigkeit und Benutzerfreundlichkeit überzeugt. Um dies zu erreichen, haben wir ein strukturiertes Vorgehen gewählt, das systematische Testverfahren, die Identifizierung und Behebung von Fehlern sowie die gezielte Implementierung neuer Features umfasst.

Insgesamt wurden während der gesamten Entwicklungsperiode **114 Issues** identifiziert, die über das Issue-Tracking von GitHub erfasst, klassifiziert und nach ihrer Priorität eingestuft wurden. Im Folgenden sind alle nach der Implementierungsphase erfassten und behobenen Issues thematisch gruppiert aufgeführt.

2 Bugfixing

2.1 Frontend

2.1.1 Styling

#164 Fix Inconsistent Focus Borders on Frontend Input Fields

- **Fehlersymptom:** Die Rahmen von Auswahlfeldern waren inkonsistent, hatten unterschiedliche Stärken und fehlten teilweise.
- **Ursache des Fehlers:** Die Rahmenstile waren bei den verschiedenen Typen von Eingabefeldern uneinheitlich definiert.
- **Fehlerbehebung:** Das Border-Styling wurde über alle Eingabefelder hinweg vereinheitlicht, um ein konsistentes Erscheinungsbild zu gewährleisten.

#255 Fix Notification Item Layout

- **Fehlersymptom:** Das Entfernen-Icon in den Benachrichtigungen war an der falschen Position und verschoben.
- **Ursache des Fehlers:** Ein Layout-Fehler bei langen Benachrichtigungen führte zu Verschiebungen des Icons.
- **Fehlerbehebung:** Das Styling wurde korrigiert und durch die Verwendung der entsprechenden Tailwind CSS Klassen angepasst, um eine

stabile und korrekte Position des Icons unabhängig von der Benachrichtigungslänge zu gewährleisten.

2.1.2 Popups

#211 Fix and Update View User Popup

- **Fehlersymptom:** Das Layout und die Anzeige der Icons waren fehlerhaft.
- **Ursache des Fehlers:** Der Fehler trat auf, nachdem die Anwendung von der Verwendung von SVG-Bilddateien auf React-SVG-Komponenten umgestellt wurde, um die Performance zu verbessern.
- **Fehlerbehebung:** Die Benutzeransicht wurde vollständig an die neuen React-SVG-Komponenten angepasst. Dabei wurde das gesamte Layout überarbeitet und verbessert. Zusätzlich wurden neue Informationen, wie die Slack-ID des Nutzers, in die Ansicht integriert.

#256 Missing Signature Confirmation Popup on Supervisor

- **Fehlersymptom:** Beim Anklicken des **Sign Sheet**-Buttons durch den Supervisor erschien kein Bestätigungs-Popup, und das Timesheet wurde direkt signiert.
- **Ursache des Fehlers:** Das Bestätigungs-Popup wurde auf der Supervisor-Seite nicht integriert.
- **Fehlerbehebung:** Das Bestätigungs-Popup wurde nachträglich eingefügt, um eine Bestätigung vor der Signierung des Timesheets zu ermöglichen.

#257 Remove Add Project Button within Supervisor

- **Fehlersymptom:** Der **Add Project**-Button öffnete fälschlicherweise das Time-Tracking-Popup.
- **Ursache des Fehlers:** Da das **Add Project**-Feld ursprünglich als optionales Kriterium geplant war, wurde es vorübergehend mit der Time-Tracking-Funktionalität verknüpft.
- **Fehlerbehebung:** Die Popup-Funktionalität des Buttons wurde deaktiviert, um das unerwünschte Öffnen des Time-Tracking-Popups zu verhindern.

2.1.3 Timesheet

#158 Optimize High Priority Timesheet Retrieval for Hiwi

- **Fehlersymptom:** Nachdem ein Hiwi ein Timesheet signiert hatte, wurde automatisch zum Timesheet mit der nächsthöheren Priorität gewechselt. Wenn der Hiwi mehrere unsignierte Timesheets hatte, entstand der Eindruck, dass das Signieren nicht erfolgreich war, was zu Verwirrung und einer schlechten Benutzererfahrung führte.
- **Ursache des Fehlers:** Die Hiwi-Startseite ist so konzipiert, dass immer das Timesheet mit der höchsten Priorität angezeigt wird.
- **Fehlerbehebung:** Anstatt direkt nach dem Signieren das neu priorisierte Timesheet zu laden, bleibt das gerade signierte Timesheet mit dem aktualisierten Status zunächst sichtbar. Erst beim erneuten Laden der Seite oder beim Durchblättern gelangt der Nutzer zum Timesheet mit der nächsthöheren Priorität.

#159 Integrate Vacation Minutes into Timesheets View

- **Fehlersymptom:** Die Anzeige der in einem bestimmten Monat genommenen Urlaubsminuten in der Dokumentenansicht des Hiwis wurde nicht korrekt dargestellt.
- **Ursache des Fehlers:** Eine notwendige Berechnung im Backend, die die Summe der eingetragenen Urlaubsminuten ermittelt, fehlte. Zudem gab es kein entsprechendes Feld in der Datenbank, das diese Information direkt speichert und dem Frontend zur Verfügung stellt, ohne zusätzliche Anfragen zu erfordern.
- **Fehlerbehebung:** Es wurde ein neues Feld `vacationMinutes` im Timesheet eingeführt, das die Anzahl der eingetragenen Urlaubsminuten speichert. Dieses Feld wird bei der Erstellung oder Änderung von Urlaubszeiten automatisch aktualisiert, sodass die korrekten Daten direkt in der Dokumentenansicht angezeigt werden können.

#258 Cases of Wrongly Converted Negative Minutes to Hours

- **Fehlersymptom:** In einigen Fällen wurden negative Minuten um eine Stunde falsch angezeigt.
- **Ursache des Fehlers:** Rundungsfehler traten auf, weil die negative Minutenanzahl nicht korrekt berücksichtigt wurde.

- **Fehlerbehebung:** Die Rundung wurde unabhängig vom Vorzeichen durchgeführt und das Vorzeichen anschließend korrekt wieder hinzugefügt.

2.1.4 Users

#178 Show Archived Users in Admin Homepage

- **Fehlersymptom:** Archivierte User werden nicht auf der Admin Homepage angezeigt.
- **Ursache des Fehlers:** Endpoint für das Abfragen der archivierten User war noch nicht erstellt worden.
- **Fehlerbehebung:** Erweiterung des Filters um einen Archived Filter und Anbindung der Admin Homepage an den Endpoint zum Abfragen der archivierten User.

#204 Enable Hiwi Users to View Available Vacation Minutes

- **Fehlersymptom:** Hiwis konnten ihr Urlaubskonto nicht einsehen und somit nicht feststellen, wie viele Urlaubsstunden ihnen noch zur Verfügung standen.
- **Ursache des Fehlers:** Es fehlte eine Implementierung zur Anzeige des Urlaubskontos im Frontend.
- **Fehlerbehebung:** Eine Anzeige des aktuellen Urlaubskontos wurde im Add Vacation Popup integriert, sodass Hiwis nun direkt sehen können, wie viele Urlaubsstunden sie noch übrig haben.

#230 Deactivate Supervisor Loading when no Supervisor exists

- **Fehlersymptom:** Anzeigefehler bei der Hiwi-Erstellung, wenn noch kein Supervisor im System registriert ist.
- **Ursache des Fehlers:** Der Fehler wurde durch das Supervisor-Dropdown-Feld verursacht, das keine Auswahlmöglichkeit bietet, wenn noch kein Supervisor vorhanden ist. Dadurch war die Erstellung eines Hiwis nicht möglich.
- **Fehlerbehebung:** Wenn kein Supervisor im System registriert ist, wird die Hiwi-Erstellung deaktiviert. Die Hiwi-Rolle ist im Rollen-Dropdown-Menü bei der Benutzererstellung nicht mehr auswählbar, bis ein Supervisor existiert.

2.1.5 Pages

#200 Replace Static Project on Hiwi Documents Page

- **Fehlersymptom:** Ein statischer Platzhalter blieb fälschlicherweise in der Hiwi-Dokumentansicht sichtbar.
- **Ursache des Fehlers:** Der Platzhalter war ursprünglich als vorübergehende Lösung gedacht und wurde versehentlich nicht entfernt.
- **Fehlerbehebung:** Der Platzhalter wurde durch eine Anzeige der letzten Timesheet-Statusänderung ersetzt.

#241 Secretary Entry Sorting

- **Fehlersymptom:** Auf der Sekretariats-Homepage wurden auch Nutzer angezeigt, die im ausgewählten Monat noch kein Timesheet erstellt haben. Diese Nutzer erschienen unsortiert zwischen den Hiwis mit vorhandenen Timesheets, was die Übersichtlichkeit und Nachverfolgbarkeit für das Sekretariat erschwerte.
- **Ursache des Fehlers:** Es fehlte eine Sortierung im Frontend, die Hiwis basierend auf ihren Timesheets ordnet.
- **Fehlerbehebung:** Ein zusätzlicher Sortiermechanismus wurde implementiert, der sicherstellt, dass Hiwis mit einem Timesheet mit gültigem Status zuerst angezeigt werden, um die Übersichtlichkeit zu verbessern.

#136 Filters Not Preserved After User Edit

- **Fehlersymptom:** Die User-Rollen-Filter wurden zurückgesetzt, sobald ein Nutzer editiert wurde.
- **Ursache des Fehlers:** Das Schließen des Popups nach der Bearbeitung eines Nutzers führte zu einem vollständigen Neuladen der Seite, wodurch der Status des Filters verloren ging.
- **Fehlerbehebung:** Das vollständige Neuladen der Seite wurde verhindert. Stattdessen werden nur die aktualisierten Nutzerdaten neu geladen, sodass die Filtereinstellungen beibehalten werden und die Nutzerliste weiterhin konsistent bleibt.

2.1.6 Input and Error Handling

#199 Enhance Decimal Input Handling

- **Fehlersymptom:** Bei der Eingabe von Dezimalzahlen blieb die vor-eingestellte Null stehen, was die Eingabe der gewünschten Dezimalzahl erschwerte. Zudem war der Nutzer auf die Komma-Notation beschränkt.
- **Ursache des Fehlers:** Die Eingabefelder waren noch nicht vollständig auf die Verarbeitung von Dezimalzahlen optimiert, was zu Problemen bei der Benutzerfreundlichkeit führte.
- **Fehlerbehebung:** Die Eingabefelder wurden überarbeitet, um eine optimale Unterstützung für Dezimalzahlen zu gewährleisten. Wenn das Feld leer ist, wird eine 0 angezeigt. Sobald der Benutzer mit der Eingabe beginnt, wird die 0 durch die eingegebene Zahl ersetzt. Zudem wurde die Unterstützung für sowohl Punkt- als auch Komma-Notation hinzugefügt.

#160 Implement Error Handling for Document Downloads

- **Fehlersymptom:** Beim Versuch, Dokumente herunterzuladen, wurden keine korrekten Fehlermeldungen angezeigt, wenn der Download fehlschlug.
- **Ursache des Fehlers:** Der Fehler lag in der Art der Anfrage an das Backend. Da es sich um einen Filestream handelte, wurde die JSON-Antwort des Backends nicht gleichzeitig empfangen.
- **Fehlerbehebung:** Es wurde ein Mechanismus implementiert, der bei einem fehlgeschlagenen Download die Antwort des Backend-Endpunkts abfragt und auswertet, sodass eine korrekte Fehlermeldung angezeigt wird.

#229 Allow Leading Zero for Personal Number Input Field

- **Fehlersymptom:** Es war nicht möglich, Personalnummern mit führenden Nullen einzugeben, was die Flexibilität bei der Eingabe einschränkte.
- **Ursache des Fehlers:** Führende Nullen wurden automatisch überschrieben, sobald eine Zahl in das Eingabefeld eingegeben wurde, wodurch die Eingabe von Nummern mit führenden Nullen verhindert wurde.
- **Fehlerbehebung:** Die Validierung des Eingabefeldes wurde angepasst, sodass nun auch führende Nullen bei der Eingabe erlaubt sind.

2.1.7 Further Issues

#151 Add Overtime to Total Working Hours in Frontend

- **Fehlersymptom:** Es gab keine Möglichkeit für Hiwis, ihre aktuellen Überstunden einzusehen.
- **Ursache des Fehlers:** Eine entsprechende Anzeige für die Überstunden fehlte.
- **Fehlerbehebung:** Eine neue Anzeige für die aktuellen Überstunden des Hiwis wurde in die Progress Bar Card integriert.

2.2 Backend

2.2.1 Document Generation

#99 Mark Vacation Entries in Timesheets

- **Fehlersymptom:** Beim Erstellen eines Timesheet-PDFs blieb das Feld "Tätigkeit" bei Urlaubseinträgen leer.
- **Ursache des Fehlers:** Bei Urlaubs-Einträgen wird kein Feld Tätigkeit gespeichert, also wurde dies bei der PDF Erstellung ignoriert.
- **Fehlerbehebung:** Für mehr Überschaubarkeit, wird bei Urlaubseinträgen im PDF Urlaub als Tätigkeit eingetragen.

#217 Document Generation Inaccurate Fields

- **Fehlersymptom:** Beim Erstellen eines Timesheet-Dokuments wurde bei den Unterschriften als Datum das Datum der Dokumenten-Erstellung angezeigt.
- **Ursache des Fehlers:** Beim Erstellen des PDFs wurde als Datum das aktuelle Datum eingetragen.
- **Fehlerbehebung:** Statt dem aktuellen Datum wird das Datum der letzten Status-Änderung des Timesheets eingetragen.

#192 Correct Timezone Handling for Timestamps in Database

- **Fehlersymptom:** Timestamps wurden in der Datenbank mit UTC-Zeiten gespeichert, aber fälschlicherweise mit deutscher Ortszeit verknüpft. Dies führte zu einer Inkonsistenz, bei der im Frontend die Benachrichtigungen um zwei Stunden verschoben angezeigt wurden.

- **Ursache des Fehlers:** Die Zeiten wurden fälschlicherweise in deutscher Ortszeit als UTC gespeichert, was zu einer Verschiebung um die Stunden der Zeitzonendifferenz führte.
- **Fehlerbehebung:** Alle Zeiten werden nun einheitlich in UTC in der Datenbank gespeichert. Die Umrechnung in die lokale Zeit erfolgt ausschließlich im Frontend und bei der Erstellung von Dokumenten, wodurch eine konsistente und korrekte Anzeige gewährleistet ist.

#220 Write Slack Access Token in DB

- **Fehlersymptom:** Der Slack Access Token war im Code fest hinterlegt, was ein Sicherheitsrisiko darstellte.
- **Ursache des Fehlers:** Der Token wurde provisorisch für Testzwecke direkt im Code hinterlegt.
- **Fehlerbehebung:** Eine neue **Administration**-Collection wurde erstellt, in der der Access Token sicher gespeichert und bei Bedarf abgefragt wird. Dadurch kann der Token nur von privilegierten Admins mit Datenbankzugriff ausgelesen werden.

2.2.2 Timesheet

#162 Adjust Endpoint to Return Latest Completed Timesheet

- **Fehlersymptom:** Falls alle Timesheets Complete waren, wurde das älteste Timesheet zurück gegeben. In diesem Fall hätte aber das neuste Timesheet die höchste Priorität.
- **Ursache des Fehlers:** Die Methode `getHighestPriorityTimesheet` ordnete die Timesheets zuerst nach Status (wobei Complete die niedrigste Priorität hatte), und dann absteigend nach Datum. So hätte zum Beispiel das älteste Timesheet mit Revision Status die höchste Priorität. Falls alle Timesheets Complete waren, wurde daher nur nach absteigendem Datum sortiert.
- **Fehlerbehebung:** Es wird geprüft, ob das Timesheet mit der höchsten Priorität Complete ist. In diesem Fall sind alle Timesheets Complete und es wird stattdessen das neuste zurückgegeben.

#131 Add vacationMinutes Field to Timesheets

- **Fehlersymptom:** Im Frontend konnte nicht die genommene Urlaubszeit pro Timesheet angezeigt werden.
- **Ursache des Fehlers:** Es wurde im Timesheet nicht gespeichert, wie viel Urlaub schon genommen wurde.
- **Fehlerbehebung:** Beim Ändern des Timesheet wird jetzt nicht nur die gesamte Arbeitszeit, sondern auch Urlaubszeit berechnet, und im Timesheet gespeichert.

#191 Clarify Signing Order for Multiple Timesheets

- **Fehlersymptom:** Timesheets konnten in beliebiger Reihenfolge signiert werden. Dies konnte dazu führen, dass zum Beispiel die Überstunden nicht richtig berechnet wurden.
- **Ursache des Fehlers:** Beim Signieren eines Timesheets wurde nicht geprüft, ob alle anderen Timesheets, welche weiter in der Vergangenheit lagen, schon Complete waren. Dadurch konnten Änderungen bei einem schon signierten Timesheet entstehen.
- **Fehlerbehebung:** Es wurde eine neue Timesheet-Validation Strategie implementiert, welche beim Signieren sicherstellt, dass alle vorherigen Timesheets schon Complete sind und es somit zu keinen zukünftigen Änderungen des signierten Timesheets kommt.

#179 Ensure Validation for Timesheet and TimeEntry

- **Fehlersymptom:** Bei der Erstellung von Timesheets oder Time Entries wurde nicht vollständig überprüft, ob die Daten korrekt und zulässig sind. Dadurch konnte zum Beispiel ein Timesheet signiert werden, obwohl das Timesheet des vorherigen Monats noch in Bearbeitung war.
- **Ursache des Fehlers:** Obwohl Validierungsklassen implementiert waren, wurden sie in den Service-Klassen nicht korrekt aufgerufen. Zudem fehlten einige Validierungsstrategien, um bestimmte Szenarien abzudecken.
- **Fehlerbehebung:** Die Validierung wird nun korrekt im Timesheet-Service und Time-Entry-Service aufgerufen. Außerdem wurde eine neue Timesheet-Strategie hinzugefügt, die beim Signieren sicherstellt, dass kein vorheriges Timesheet noch in Bearbeitung ist.

#76 Update Last Signature on Timesheet Status Change

- **Fehlersymptom:** Beim Erstellen eines Timesheet-Dokuments wurde das aktuelle Datum als Signaturdatum eingetragen, anstatt des tatsächlichen Datums der Signatur.
- **Ursache des Fehlers:** Im Dokumentendatenmodell wurde das aktuelle Datum verwendet, da zum Zeitpunkt der Dokumentenerstellung das tatsächliche Signaturdatum noch nicht gespeichert war.
- **Fehlerbehebung:** Bei jeder Statusänderung eines Timesheets wird nun das Datum der letzten Änderung gespeichert. Dieses Datum wird dann verwendet, um das korrekte Signaturdatum im Dokument einzutragen.

#232 Lack of Validation for Timesheet Creation Order

- **Fehlersymptom:** Hiwis konnten Timesheets für zurückliegende Monate erstellen und dadurch ihre verfügbare Urlaubszeit unbegrenzt erhöhen.
- **Ursache des Fehlers:** Bei der Erstellung eines Timesheets wurden automatisch Urlaubsstunden gutgeschrieben, ohne eine Einschränkung auf den Zeitraum, in dem der Hiwi Timesheets erstellen darf.
- **Fehlerbehebung:** Die Timesheet-Erstellung wurde auf Monate nach der Account-Erstellung des Hiwis beschränkt. Dadurch kann der Hiwi keine Timesheets für Zeiträume vor seiner Einstellung erstellen.

#243 Sign Sheet Validation

- **Fehlersymptom:** Hiwis konnten Timesheets signieren, obwohl zu wenig Arbeitsstunden erfasst waren.
- **Ursache des Fehlers:** Beim Signieren eines Timesheets wurde nicht geprüft, wie viele Arbeitsstunden eingetragen wurden.
- **Fehlerbehebung:** Beim Signieren wird geprüft, ob mindestens 80% der Arbeitszeit (nach Vertrag) gearbeitet wurden. Wenn dies nicht der Fall ist, kann nicht signiert werden.

#253 Implement Restrictions on Future Timesheet Creation

- **Fehlersymptom:** Timesheets konnten für Monate in der Zukunft erstellt werden. Dadurch wurde die verfügbare Urlaubszeit erhöht, wodurch sie im Frontend nicht korrekt angezeigt werden konnte.
- **Ursache des Fehlers:** Es gab keine Einschränkung für das Erstellen von Timesheets in der Zukunft.
- **Fehlerbehebung:** Das Erstellen von Timesheets wird auf Monate bis zum aktuellen Monat eingeschränkt.

2.2.3 Users

#132 Endpoint for Accessing Contractual Data of Assigned Hiwis

- **Fehlersymptom:** Im Frontend konnte kein Diagramm angezeigt werden, welches die gearbeiteten Stunden des Hiwis, mit den Soll-Arbeitsstunden des Vertrags vergleicht.
- **Ursache des Fehlers:** Es gab keinen Endpoint für den Supervisor, welcher die Vertragsinformationen anderer Nutzer an das Frontend übermitteln konnte.
- **Fehlerbehebung:** Es wurde ein neuer Endpoint implementiert, welcher die Vertragsinformationen zu einem gegebenen Hiwi ausgibt.

#117 Ensure Complete Deletion of User Data

- **Fehlersymptom:** Es blieben Daten von bereits gelöschten Nutzern in der Datenbank bestehen. Wenn ein neuer Nutzer mit dem gleichen Benutzernamen erstellt wurde, konnten fälschlicherweise die Daten des zuvor gelöschten Nutzers diesem neuen Nutzer zugeordnet werden.
- **Ursache des Fehlers:** Beim Löschen eines Nutzers wurden die zugehörigen Timesheets, Time-Entries und Dateien nicht aus der Datenbank entfernt.
- **Fehlerbehebung:** Beim Löschen eines Nutzers werden nun alle zugehörigen Daten vollständig aus der Datenbank gelöscht. Vor dem Löschen eines Supervisors wird zudem überprüft, ob diesem noch Hiwis zugeordnet sind, um eine vollständige Datenkonsistenz zu gewährleisten.

#85 Remove Timesheets-Field From Hiwi

- **Fehlersymptom:** Es existierten nicht genutzte Felder in den Hiwi-Datensätzen, die auf eine veraltete Systemarchitektur und frühere Datenbankstrukturen zurückgingen.
- **Ursache des Fehlers:** Diese Felder waren Überbleibsel aus einer früheren Version des Systems, deren Architektur seitdem überarbeitet wurde, aber der entsprechende Code war nicht vollständig bereinigt worden.
- **Fehlerbehebung:** Das ungenutzte Timesheets-Feld wurde vollständig aus den Hiwi-Datensätzen und dem zugehörigen Code entfernt, sodass es nicht länger in die Datenbank geschrieben wird. Dies reduziert die Komplexität und stellt sicher, dass nur relevante Daten gespeichert werden.

#80 Safety Checks to Prevent Admins from Deleting Themselves

- **Fehlersymptom:** Admins hatten die Möglichkeit, sich selbst zu löschen. Dadurch bestand das Risiko, dass ein Admin alle anderen Admins löschen und sich im schlimmsten Fall selbst aus dem System ausschließen konnte.
- **Ursache des Fehlers:** Es fehlte eine Validierung, die verhinderte, dass ein Nutzer seinen eigenen Account löschen kann.
- **Fehlerbehebung:** Eine Validierungsmaßnahme wurde implementiert, die sicherstellt, dass ein Admin sich nicht selbst löschen kann.

#34 Extending the Validation to Validate Contract Information

- **Fehlersymptom:** Die Validierung der Hiwi-Vertragsinformationen wurde im Backend nicht durchgeführt.
- **Ursache des Fehlers:** Es fehlten notwendige Validierungsmaßnahmen, die im Backend implementiert werden sollten.
- **Fehlerbehebung:** Einführung zusätzlicher Validierungsmechanismen im Backend, die die Gültigkeit der Felder `monthlyWorkingHours` und `hourlyWage` sicherstellen.

#248 Prevent Archiving of Default Admin

- **Fehlersymptom:** Admin User konnten archiviert werden.
- **Ursache des Fehlers:** Es fehlte eine Validierung, die das Archivieren von Admin-Accounts verhindert.
- **Fehlerbehebung:** Eine zusätzliche Validierungsmaßnahme wurde implementiert, die überprüft, ob der zu archivierende Benutzer ein Admin ist und das Archivieren in diesem Fall unterbindet.

#249 Fix Account Creation Date in User Profile

- **Fehlersymptom:** Der Timestamp für die Accounterstellung entsprach nicht dem tatsächlichen Erstellungszeitpunkt und war identisch für mehrere Benutzer.
- **Ursache des Fehlers:** Der Erstellungszeitpunkt wurde als Default-Wert im Konstruktor definiert. Da diese Default-Werte beim Laden der Klasse und nicht bei der tatsächlichen Erstellung des Accounts gesetzt wurden, erhielten alle Benutzer nach dem Start denselben Timestamp.
- **Fehlerbehebung:** Der Erstellungszeitpunkt wurde korrekt innerhalb der User-Klasse beim tatsächlichen Erstellen des Accounts gesetzt.

2.2.4 Time Entry

#205 Enforce Vacation Minutes Limits for Hiwi Users

- **Fehlersymptom:** Hiwis konnten mehr Urlaub nehmen, als auf ihrem Urlaubskonto verfügbar war.
- **Ursache des Fehlers:** Es fehlte eine Validierung, die sicherstellt, dass das Urlaubskonto nicht ins Negative fallen kann.
- **Fehlerbehebung:** Eine zusätzliche Validierung wurde implementiert, die überprüft, dass das Urlaubskonto stets positiv bleibt und nur Urlaub eingetragen werden kann, wenn genügend Minuten verfügbar sind.

#212 Break of 15min Is Always Required

- **Fehlersymptom:** Es war nicht möglich, Zeiteinträge ohne Pausen hinzuzufügen, obwohl dies bei Arbeitsphasen unter 6 Stunden erlaubt sein sollte.

- **Ursache des Fehlers:** Die Validierungsstrategie für Pausen bei Time-Entries verlangte fälschlicherweise eine Pause von 15 Minuten bei Arbeitsphasen bis zu 6 Stunden.
- **Fehlerbehebung:** Die erforderliche Pausenlänge für Arbeitszeiten bis 6 Stunden wurde auf 0 Minuten reduziert.

#235 Implement Time Entry Validation to Exclude Weekend

- **Fehlersymptom:** Es war möglich, Zeiteinträge an Wochenenden zu erstellen.
- **Ursache des Fehlers:** Eine vorhandene Validierungsstrategie prüfte lediglich, ob Zeiteinträge an Feiertagen erstellt wurden, jedoch nicht, ob diese an Wochenenden stattfanden.
- **Fehlerbehebung:** Es wurde eine neue Validierungsstrategie implementiert, die sicherstellt, dass Zeiteinträge nicht an Wochenenden erstellt werden können.

#252 Restrict Time Entry Working Hours

- **Fehlersymptom:** Arbeitszeiteinträge, die mitten in der Nacht liegen, wurden nicht verhindert.
- **Ursache des Fehlers:** Es gab zwar einen Validierungsmechanismus, der Arbeitszeiteinträge außerhalb der regulären institutsinternen Geschäftszeiten mit einer Warnung markierte, jedoch bestand für den Nutzer keine Einschränkung bei der Eingabe solcher Zeiten.
- **Fehlerbehebung:** Ein zusätzlicher Validierungsmechanismus wurde implementiert. Wenn die eingetragenen Zeiten insgesamt mehr als 2 Stunden von den regulären Geschäftszeiten abweichen, wird der Eintrag als ungültig markiert und abgelehnt. Bei geringeren Abweichungen erfolgt eine Validierung mit einer Warnung.

#251 Restrict Multiple Time Entries on the Same Date

- **Fehlersymptom:** Es war möglich, mehrere Urlaubs- und Arbeitszeiteinträge für denselben Tag zu erstellen.
- **Ursache des Fehlers:** Es fehlte eine Validierung, die das Erstellen mehrerer Einträge am selben Tag verhindert.
- **Fehlerbehebung:** Implementierung einer Validierung, die prüft, ob bereits Einträge für den ausgewählten Tag existieren, um doppelte Einträge zu verhindern.

2.2.5 DB Setup

#221 Create DB Setup Method

- **Fehlersymptom:** Beim ersten Start der Anwendung war es keinem Nutzer möglich, sich anzumelden.
- **Ursache des Fehlers:** Beim initialen Start der Anwendung war kein Administrator im System registriert. Dadurch war eine Anmeldung für alle Nutzer unmöglich, was die Nutzung der Anwendung verhinderte.
- **Fehlerbehebung:** Eine Setup-Routine wurde implementiert, die beim ersten Start alle notwendigen Datenbank-Collections initialisiert und automatisch einen Standard-Administrator erstellt, falls noch keiner vorhanden ist. Dadurch wird sichergestellt, dass die Anwendung von Anfang an funktionsfähig ist.

#246 Setup Service - Fix Multiprocessing Issue

- **Fehlersymptom:** Bei der Parallelisierung des Flask-Backends innerhalb des Docker-Containers kam es zu mehrfachen Aufrufen der Setup-Methoden, was zur Erstellung mehrerer Default-Admin-User führte.
- **Ursache des Fehlers:** Es fehlte eine Synchronisationsmaßnahme, um den kritischen Abschnitt vor gleichzeitigen Zugriffen zu schützen.
- **Fehlerbehebung:** Eine Synchronisationsmaßnahme wurde implementiert, um den gleichzeitigen Zugriff auf den kritischen Abschnitt zu verhindern und so das mehrfache Ausführen des Setups zu unterbinden.

2.2.6 Further Issues

#231 Intermittent Errors

- **Fehlersymptom:** Beim Erstellen neuer Timesheets, die noch nicht im System existierten, trat ein Fehler auf, der Folgeanfragen verhinderte.
- **Ursache des Fehlers:** Der Fehler entstand durch die unvollständige Entfernung veralteter Datenstrukturen, bei der versucht wurde, Object IDs zu serialisieren (jsonify), was fehlschlug.
- **Fehlerbehebung:** Die vollständige Entfernung der veralteten Timesheet-Datenstrukturen wurde abgeschlossen, wodurch das Problem behoben wurde.

#180 Allow Usage of Specific Special Chars in Names

- **Fehlersymptom:** Nutzer konnten keine Sonderzeichen in ihren Namen eingeben, sodass Namen wie „René“ nicht akzeptiert wurden.
- **Ursache des Fehlers:** Die Validierung erlaubte nur Namen, die aus den Buchstaben A-Z bestanden.
- **Fehlerbehebung:** Die Regex-Validierungsregeln wurden so angepasst, dass nun alle Unicode-Zeichen akzeptiert werden.

3 Frontend Testing Routine

Dieses Kapitel beschreibt die systematischen Testverfahren, die zur Validierung der Frontend-Komponenten unseres Systems durchgeführt wurden. Die Tests decken verschiedene Anwendungsfälle ab und gewährleisten eine umfassende Überprüfung aller relevanten Funktionen, Benutzeroberflächen und Datenverarbeitungen. Ziel war es, die Korrektheit, Benutzerfreundlichkeit und Robustheit des Frontends sicherzustellen.

3.1 Initialisierung und Setup-Prüfung

- **Anmeldung mit Default-User irladmin:** Überprüfung, ob die Anmeldung korrekt funktioniert und ob nach der ersten Anmeldung die notwendigen Initialisierungen korrekt ablaufen. Dazu gehört die Erstellung der **Administration**-Collection mit einem leeren Slack-Token sowie die Erstellung eines Default-Admin-Benutzers, falls keiner existiert.
- **Passwort zurücksetzen:** Prüfung des Prozesses zum Zurücksetzen des Passworts und der erfolgreichen Anmeldung mit dem neuen Passwort.
- **Abmelden und Neu-Anmelden:** Sicherstellung, dass der Benutzer nach dem Abmelden korrekt abgemeldet wird und keine alten Sitzungsdaten erhalten bleiben. Nach der Neuansmeldung wurden die Benutzerrechte und -daten verifiziert.

3.2 Benutzerverwaltung: FrontendSupervisor

Bei der Erstellung und Verwaltung eines **FrontendSupervisors** wurden mehrere Aspekte überprüft:

- **Erstellung eines neuen Supervisors:** Prüfung der Eingabevalidierung, z.B. bei E-Mail-Adressen (korrektes Format).
- **Archivierung und Aktivierung:** Test der Archivierung und anschließenden Aktivierung eines Supervisors. Überprüfung, ob alle zugewiesenen Hiwis korrekt aktualisiert werden.
- **Login und Logout:** Sicherstellung der ordnungsgemäßen Verwaltung der Sitzungen nach Aktivierung.

3.3 Benutzerverwaltung: FrontendHiwi

Zwei Hiwi-User (FrontendHiwi und FrontendHiwi2) wurden erstellt und getestet:

- **Erstellung und Validierung der Dezimal-Eingabefelder:** Prüfung der korrekten Eingabe und Speicherung von Dezimalzahlen, z.B. 14.5 Euro.
- **Zuweisung des Betreuers FrontendSupervisor:** Überprüfung der korrekten Zuordnung und Anzeige im UI sowie Konsistenz der Daten in der Datenbank.
- **Vergleich des Erstellungsdatums:** Abgleich zwischen dem tatsächlichen Erstellungsdatum und dem Datenbank-Timestamp, z.B. 14:25 Uhr vs. 2024-08-21T12:25:0.000+00:00 - Unter Berücksichtigung der Zeitzone.

3.4 Archivierung und Reaktivierung von Benutzern

Die Archivierungsfunktion wurde intensiv getestet, insbesondere in Kombination mit der Zuweisung von Hiwis an Supervisoren:

- **Archivierung eines Supervisors mit zugewiesenen Hiwis:** Sicherstellung, dass die Archivierung blockiert wird, wenn noch Hiwis zugewiesen sind.
- **Erfolgreiche Archivierung und Reaktivierung von Hiwis:** Test der Entfernung eines archivierten Hiwi-Users aus der Supervisor-Hiwi-Liste und automatische Aktivierung des Supervisors bei Reaktivierung des Hiwi.

3.5 Funktionsprüfung: FrontendHiwi

Die Funktionalitäten eines Hiwi-Accounts wurden im Detail überprüft:

- **Prüfung der Vertragsseite:** Kontrolle der korrekten Darstellung von Vertragsinformationen und Überprüfung, ob Seiten wie `Home` und `Documents-Ansicht` initial leer sind.
- **Urlaubseinträge:** Test, ob die Erstellung eines Urlaubseintrags korrekt verhindert wird, wenn kein Urlaub verfügbar ist.
- **Arbeitseinträge in unzulässigen Zeiträumen:** Sicherstellung, dass Einträge vor der Accounterstellung oder in zukünftigen Monaten mit entsprechenden Fehlermeldungen abgelehnt werden.
- **Datenvalidierung bei Arbeitseinträgen:**
 - Ablehnung von Arbeitszeiten zwischen 0 und 6 Uhr mit Fehlermeldung
 - Verhinderung von Arbeitsspannen über 10 Stunden mit Fehlermeldung
 - Prüfung der Pausenregelung: 30 Minuten Pause nach 6 Stunden, 45 Minuten nach 9 Stunden; mit Fehlermeldung.
 - Ablehnung von Arbeitseinträgen an Wochenenden und Feiertagen mit spezifischen Fehlermeldungen, inklusive Name des Feiertags z.B.: „Tag der Deutschen Einheit“ bei Erstellung am 03.10.
- **Bearbeitung von Arbeitseinträgen:** Sicherstellung, dass die Validierungsregeln auch bei Änderungen greifen.
- **Signatur des Timesheets:** Überprüfung, ob eine Fehlermeldung erscheint, wenn die Signatur noch nicht hinterlegt ist. Nach Upload der Signatur wurde die korrekte Speicherung in der Datenbank geprüft. Erneute Prüfung der Signatur nach Erstellung von Zeiteinträgen bis zu 68 Stunden (mehr als 80% der Vertragsstunden und somit zulässig).
- **Prüfung der Dokumentansicht nach Abschluss eines Timesheets:**
 - Überprüfung, ob das Dokument nach Erreichen des Status `Complete` erfolgreich heruntergeladen werden kann.
 - Prüfung, ob eine fehlende Signatur zum Fehlerfall beim Download führt und der Benutzer darauf hingewiesen wird.

- Sicherstellung, dass die PDF-Dokumentation der Arbeitszeiten korrekt mit allen Timesheet-Daten gefüllt wird, inklusive der korrekten Darstellung der Unterschriften beider beteiligten Parteien.
- Sicherstellung, dass der Download-Button nach Erreichen des Status **Complete** anklickbar ist.
- Test der Filterfunktion nach Timesheet-Status.
- Korrekte Darstellung und Anzeige der Timesheet-Informationen (Arbeitszeit, Urlaubszeit, Überstunden) im Abgleich mit den Time Entry Daten der jeweiligen Monate.

3.6 Funktionsprüfung: FrontendSupervisor

Die Funktionalität eines Supervisor-Accounts wurde umfassend getestet, insbesondere im Hinblick auf:

- **Anzeige und Verwaltung der zugewiesenen Hiwis:** Kontrolle der korrekten Anzeige und des Status der Timesheets.
- **Notification Bar:** Test der korrekten Anzeige von Benachrichtigungen über signierte Timesheets und deren Entfernung nach Löschung der Benachrichtigung.
- **Bearbeitung eines signierten Timesheets:** Überprüfung, ob nach Hochladen einer Signatur diese korrekt gespeichert und angezeigt wird. Test der Anforderung von Änderungen und Statusänderung auf „Revision“.
- **Prüfung der Employees Page und Signatur:**
 - Hochladen der Unterschrift per Drag & Drop.
 - Sicherstellung, dass die Signatur korrekt dargestellt und in der Datenbank gespeichert wird.
- **Änderung beim signierten Timesheet anfordern:**
 - Definition eines Änderungstextes.
 - Prüfung, ob der Status des Timesheets auf „Revision“ geändert wird.
 - Sicherstellung, dass die Änderungsnachricht korrekt in den Benachrichtigungen des betroffenen Hiwi angezeigt wird.

- **Finale Signatur nach Revision:** Sicherstellung, dass nach erneuter Signatur durch den Hiwi das Timesheet final signiert und der Status auf „Complete“ gesetzt wird.

3.7 Funktionsprüfung: FrontendSecretary

Zuletzt wurde die Rolle **FrontendSecretary** getestet, wobei der Fokus auf der Verwaltung der Hiwis und der Dokumente lag:

- **Prüfung der Home-Ansicht:** Kontrolle der korrekten Anzeige aller Hiwis, der Statusdarstellung und der Funktionalität der Monatsanzeige und Filterleiste.
- **Dokumentenansicht:** Sicherstellung der korrekten Anzeige von Arbeitszeit, Urlaubsstunden, Überstunden und des Status der Timesheets.
- **Download-Funktionalität:** Überprüfung des korrekten Downloads einzelner Timesheets sowie der „Download All“-Funktion als ZIP-Archiv.

3.8 Überstundenberechnung und Übertrag

Die korrekte Berechnung und der Übertrag von Überstunden wurden ebenfalls getestet. Hierbei wurde überprüft, ob:

- **Überstunden im Monat „Complete“:** Test eines Monats mit 72 von 80 Stunden (entspricht -8 Überstunden) und Kontrolle der korrekten Übertragung der Überstunden in den Folgemonat.
- **Prüfung der Hiwi-Dokumentenansicht:** Überprüfung der korrekten Anzeige und Übertragung der Überstunden in beiden Timesheets.

4 New Features

4.1 User-Archivierung

Vor der Einführung dieses Features war es nur möglich, Nutzer vollständig zu löschen, nicht jedoch, sie zu archivieren. Mit der neuen Archivierungsfunktion können Nutzer nun deaktiviert werden, ohne ihre Daten endgültig zu löschen.

Beim Archivieren eines Nutzers werden all seine Daten verborgen, und er ist im Frontend nur über die Schnittstelle `getArchivedUsers` auffindbar. Über

diese Schnittstelle kann der Nutzer bei Bedarf wieder aktiviert werden. Im Fall von Hiwis werden diese bei einer Reaktivierung erneut ihrem vorherigen Betreuer zugewiesen. Betreuer, denen noch aktive Hiwis zugeordnet sind, können nicht archiviert werden..

4.2 Einführung von Activity Type

Mit der Einführung des Feldes **Activity Type** wurde ein neues Feature implementiert, das die Klassifizierung von Arbeitseinträgen in der Anwendung ermöglicht. Dieses Feld gibt den Aktivitätstyp eines Arbeitseintrages an und kann die Werte *Projektbesprechung* und *Projektarbeit* annehmen.

Einführung eines Dropdown-Menüs: Im Track Time Popup innerhalb des Frontends wurde ein Dropdown-Menü hinzugefügt, in dem der Nutzer den Aktivitätentyp für jeden Zeiteintrag auswählen muss. Dies gewährleistet, dass jeder Arbeitseintrag klar einem spezifischen Typ zugeordnet wird.

Anpassung der Dokumentengenerierung: Im Zuge der Implementierung wurde auch die Dokumentengenerierung angepasst. Anstatt der genauen Beschreibung der Aktivität wird in den generierten Dokumenten nun nur noch der Aktivitätentyp angezeigt, während die detaillierte Beschreibung der Aktivität weiterhin für Hiwis und Supervisoren in der Anwendung einsehbar bleibt.

Erweiterung der Datenbankstruktur: Zur Unterstützung dieses neuen Features wurde ein zusätzliches Feld für den Aktivitätentyp in der Datenbankstruktur jedes Arbeitseintrags eingeführt. Entsprechende Anpassungen im Backend wurden vorgenommen, um die Speicherung und Verarbeitung dieses neuen Feldes zu gewährleisten.

4.3 Notification Feature

Das Notification Feature wurde implementiert, um eine effiziente und benutzerfreundliche Benachrichtigungslösung innerhalb der Anwendung zu bieten. Dieses Feature besteht aus zwei Hauptkomponenten: In-App Benachrichtigungen und einer Slack-Integration.

4.3.1 In-App Benachrichtigungen

Die In-App Benachrichtigungen ermöglichen es den Nutzern, relevante Informationen direkt innerhalb der Anwendung zu erhalten. Die Benachrichtigungen werden in einer speziell dafür vorgesehenen Datenbank gespeichert, die

zwei Haupttypen von Nachrichten unterscheidet: *Timesheet Status Change* und *Reminder*.

Timesheet Status Change: Dieser Nachrichtentyp informiert Nutzer über Änderungen am Status eines Timesheets. Beispielsweise wird ein Supervisor benachrichtigt, wenn ein Hiwi sein Timesheet signiert hat. Ebenso erhält der Hiwi eine Benachrichtigung, wenn der Supervisor das Timesheet genehmigt oder Änderungen anfordert. Diese Statusänderungen werden als Nachrichten im Benachrichtigungssystem angezeigt und enthalten detaillierte Informationen über die Art der Änderung.

Reminder: Reminders sind Nachrichten, die dazu dienen, Nutzer an wichtige Ereignisse oder Aufgaben zu erinnern. Ein typisches Beispiel ist eine Erinnerung an den Nutzer, sein Timesheet zu signieren, falls der neue Monat bereits begonnen hat und das Timesheet noch nicht abgeschlossen wurde.

Benachrichtigungsanzeige: Alle Benachrichtigungen werden in der Anwendung über ein Dropdown-Menü angezeigt. Die Nachrichten sind nach Aktualität sortiert, wobei die neuesten Benachrichtigungen oben angezeigt werden. Eine Glocke auf dem Notification-Button zeigt an, ob es neue ungelesene Nachrichten gibt, und sorgt dafür, dass Nutzer keine wichtigen Informationen verpassen.

4.3.2 Slack Integration

Zusätzlich zur In-App Benachrichtigung wurde eine Slack-Integration implementiert, die es ermöglicht, Benachrichtigungen direkt über Slack an die Nutzer zu senden. Jeder Nutzer hat eine eindeutige Slack-ID, die in der Datenbank der Anwendung gespeichert ist. Über diese ID kann Clockwise Nachrichten direkt an die Slack-Accounts der Nutzer senden, sodass diese auch außerhalb der Anwendung auf dem Laufenden bleiben.

Die Slack-Integration erweitert die Reichweite der Benachrichtigungen und stellt sicher, dass wichtige Informationen auch dann schnell und zuverlässig kommuniziert werden, wenn die Nutzer nicht aktiv in der Anwendung arbeiten. Dies ist besonders nützlich für zeitkritische Updates wie Timesheet-Änderungen oder Erinnerungen, die eine sofortige Aufmerksamkeit erfordern.

4.4 Dynamic Profile Picture Hashing

Das Feature `Dynamic Profile Picture` wurde eingeführt, um das Hochladen und Abrufen von Profilbildern obsolet zu machen und gleichzeitig eine konsistente und leicht erkennbare Darstellung der Nutzer innerhalb der Anwendung zu gewährleisten. Anstelle eines hochgeladenen Profilbildes wird jedem Nutzer automatisch ein individuelles Profilbild generiert.

Funktionsweise: Jedes Profilbild besteht aus einem farbigen Hintergrund, auf dem der Anfangsbuchstabe des Vornamens des Nutzers im Vordergrund angezeigt wird. Die Hintergrundfarbe wird dabei aus einer festgelegten Farbpalette ausgewählt. Diese Auswahl erfolgt durch einen Hashing-Algorithmus, der eine Kombination aus Vor- und Nachnamen des Nutzers hasht. Dadurch wird jedem Nutzer eine eindeutige und konsistente Hintergrundfarbe zugewiesen.

Vorteile: Durch die Verwendung dieses Systems wird der Bedarf an zusätzlicher Speicherung von Profilbildern in der Datenbank vermieden. Da die Farbe dynamisch mittels Hashing generiert wird, bleibt sie konsistent. Dies führt zu einer Reduzierung des Verwaltungsaufwands und vermeidet das Problem ständig wechselnder Profilbilder. Gleichzeitig erleichtert das einheitliche Design der Profilbilder die Erkennung und Zuordnung von Nutzern innerhalb der Anwendung.

4.5 Fuzzy Search

Es wurde eine Suchfunktion in die Anwendung integriert, die es ermöglicht, nach verschiedenen Inhalten auf den Seiten zu suchen. Diese Suchfunktion bietet den Nutzern eine einfache Möglichkeit, gezielt nach Informationen zu suchen und die angezeigten Inhalte dynamisch nach dem eingegebenen Suchbegriff zu filtern.

Einführung der Fuzzy Search: Die Suchfunktion wurde als Fuzzy Search implementiert, was bedeutet, dass die Suche auch bei ungenauen oder unvollständigen Eingaben relevante Ergebnisse liefert. Der Mehrwert der Fuzzy Search liegt darin, dass sie nicht nur exakte Übereinstimmungen findet, sondern auch ähnliche Begriffe, wodurch Tippfehler oder leichte Abweichungen in der Eingabe toleriert werden.

Funktionalität der Fuzzy Search: Die Fuzzy Search ermöglicht es, auf den verschiedenen Seiten der Anwendung nach einer Vielzahl von Inhalten zu suchen. Zum Beispiel können Nutzer auf der Admin-Seite nach Namen,

Rollen, Benutzernamen und anderen Attributen suchen. Die Seite passt sich dynamisch an den eingegebenen Suchkontext an und filtert die angezeigten Ergebnisse entsprechend. Dadurch wird die Navigation und das Auffinden spezifischer Informationen erheblich erleichtert.

5 Deployment

Das Deployment unserer Software, wie im Pflichtenheft spezifiziert, erfordert eine Dockerisierung der gesamten Anwendung, um eine einfache und konsistente Bereitstellung auf dem Zielsystem, dem institutsinternen NAS, zu ermöglichen. Die Anwendung ist in drei Hauptkomponenten unterteilt: ein Frontend in React, ein Flask-Backend, das als REST-API dient, und eine MongoDB-Datenbank. Jede dieser Komponenten wird in einem eigenen Docker-Container ausgeführt, was eine klare Trennung und eine bessere Verwaltung der Ressourcen ermöglicht.

5.1 Dockerisierung der Anwendung

Um die Anwendung zu dockerisieren, mussten zunächst für jede Komponente Docker-Images erstellt werden. Diese Images dienen als Grundlage für die Container, die die einzelnen Teile der Anwendung ausführen. In den Verzeichnissen `frontend` und `backend` des GitHub-Repository befindet sich jeweils eine README-Datei, in der die nötigen Codeänderungen zur Imageerstellung aufgeführt sind. Der Prozess zur Erstellung der Docker-Images umfasst die folgende Schritte:

- **Frontend (React):** Für das Frontend, das in React entwickelt wurde, haben wir ein Docker-Image erstellt, das die Anwendung mittels Nginx bereitstellt. Die Anwendung wird zunächst innerhalb des Containers gebaut, wobei alle notwendigen Abhängigkeiten installiert und der Code für die Produktionsumgebung optimiert wird. Anschließend wird die gebaute React-Anwendung über einen Nginx-Webserver bereitgestellt.
- **Backend (Flask):** Das Flask-Backend wurde in einem separaten Docker-Image untergebracht, das eine Python-Umgebung nutzt. Zunächst werden die benötigten Abhängigkeiten anhand der `requirements.txt`-Datei installiert. Anschließend wird der gesamte Quellcode in den Container kopiert. Das Backend agiert als REST-API und wird über Flask gestartet, wobei der Container so konfiguriert ist, dass er eingehende

Anfragen über Port 5000 entgegennimmt. Das Flask-Backend verarbeitet Anfragen vom Frontend und führt Datenbankabfragen an die MongoDB durch.

- **Datenbank (MongoDB):** Für die MongoDB-Datenbank wurde ein weiteres Docker-Image verwendet, das auf der offiziellen MongoDB-Docker-Distribution basiert.

5.2 Konfiguration und Kommunikation der Container

- **Umgebungsvariablen:** Um die Abhängigkeiten zwischen den einzelnen Containern so gering wie möglich zu halten und eine gute Erweiterbarkeit und Robustheit zu erreichen verwenden wir die Umgebungsvariable "DB_HOST" zum Setzen der Datenbank-IP-Adresse beim Erstellen des Backend-Containers.

5.3 Bereitstellung auf dem Zielserver

Die fertigen Docker-Images werden schließlich in der folgenden internen Abnahme auf dem Zielserver, dem institutsinternen NAS, deployt.

6 Testing

Um die Integrität der Datenbank während der Tests sicherzustellen, haben wir in unseren Unittests spezielle `setUp` und `tearDown`-Methoden implementiert. Diese Methode sorgen dafür, dass vor und nach jedem Testlauf eine konsistente Datenbasis gewährleistet ist.

Die Testabdeckung (Test Coverage) beträgt insgesamt 91% der Zeilen.