

Thema:

**Aufbau eines Development- und IT Operations-Zyklus
sowie die Gestaltung eines Kundenportals**

Praxisprojekt

Fachbereich I - Wirtschafts- und Gesellschaftswissenschaften
Studiengang Wirtschaftsinformatik Online

Erstbetreuer: Prof. Dr.-Ing. Alexander Huber

vorgelegt von: Jan Zimmermann
920822
Niemegker Straße 18
12689 Berlin
01749916408
E-Mail: s79873@bht-berlin.de

Abgabetermin: 30. Juni 2022

Sperrvermerk

Diese Arbeit enthält vertrauliche Daten der Holzkontor Preussen GmbH. Eine Weitergabe der Arbeit im Ganzen oder in Teilen sowie das Anfertigen von Kopien (auch digital) - sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Genehmigung.

Anschrift:

Holzkontor Preussen GmbH

Heiko Schurwin

Sophienwerderweg 60

13597 Berlin

Berlin, den 30. Juni 2022

Inhaltsverzeichnis

Inhaltsverzeichnis	III
Verzeichnis der Abkürzungen und Akronyme	IV
Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
1 Einleitung.....	1
1.1 Betriebsprofil	1
1.2 Projektdefinition und Zielsetzung	3
1.3 Methodik.....	3
2 Grundlagen und Definition	4
2.1 Development- und IT Operations-Zyklus.....	4
2.2 Enterprise Resource Planning System.....	5
3 Ausgangssituation	6
4 Konzepterstellung.....	7
4.1 Entwicklungsumgebung	7
4.2 Development- und IT Operations-Zyklus.....	10
4.3 Kundenportal.....	11
5 Umsetzung	16
5.1 Entwicklungsumgebung	16
5.1.1 Webserver	17
5.1.2 integrierte Entwicklungsumgebung.....	18
5.2 Development- und IT Operations-Zyklus.....	21
5.3 Entwicklung	23
6 Projektabschluss und Fazit.....	26
A Fragebögen.....	27
B Mockups	29
C Kundenportal.....	32
Literaturverzeichnis	44

Verzeichnis der Abkürzungen und Akronyme

API	Application Programming Interface
BHT	Berliner Hochschule für Technik
CI	Continuous Integration
CD	Continuous Deployment
CSS	Cascading Style Sheets
CSV	Comma-separated values
DBMS	Datenbankmanagementsystem
DevOps	Development- und IT Operations
ERP-System	Enterprise Resource Planning System
GUI	Benutzeroberfläche
HKP	Holzkontor Preussen GmbH
HTML	Hypertext Markup Language
ID	Identifikationsnummer
IDE	integrierte Entwicklungsumgebung
OCR	Optical Character Recognition
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
SQL	Structured Query Language

Abbildungsverzeichnis

Abb. 1.1: Anlage zur Trennung/Zerkleinerung der Holzabfälle.....	1
Abb. 1.2: Holz AI zur Spanplattenherstellung geeignet.....	2
Abb. 1.3: Holz AIV nur zur Energiegewinnung geeignet	2
Abb. 1.4: branchenweit innovativer Elektro-Bagger	2
Abb. 2.1: Vergleich der Methoden – DevOps deckt den gesamten Lebenszyklus ab	4
Abb. 4.1: Anfragen pro Sekunde	8
Abb. 4.2: Speicherverbrauch	8
Abb. 4.3: Antwortzeit.....	9
Abb. 4.4: aktueller Wiegebeleg.....	13
Abb. 4.5: aktuelle Rechnung.....	14
Abb. 4.6: Mockup – Wiegungen.....	15
Abb. 5.1: Screenshot der Plugins (Strg+Alt+S).....	16
Abb. 5.2: Screenshot XAMPP	17
Abb. 5.3: Screenshot phpMyAdmin.....	18
Abb. 5.4: Terminal PhpStorm.....	18
Abb. 5.5: IDE komplett	18
Abb. 5.6: IDE – Projektübersicht.....	19
Abb. 5.7: IDE – geöffnete Datei	19
Abb. 5.8: IDE – geöffnete Datei	20
Abb. 5.9: IDE – gefundene Fehler und Probleme	20
Abb. 5.10: IDE – SonarLint.....	20
Abb. 5.11: IDE – Git Verknüpfung.....	20
Abb. 5.12: GitLab – Versionen.....	21
Abb. 5.13: GitLab – Anforderungen	21
Abb. 5.14: GitLab – Meilensteine.....	21
Abb. 5.15: GitLab – CI/CD-Pipeline	22
Abb. 5.16: Docker-Container.....	22
Abb. 5.17: TeamCity CI/CD-Pipeline.....	23
Abb. 5.18: TeamCity Auswertung	23
Abb. 5.19: Grafana Warnungen	23
Abb. 5.20: Kundenportal Lokalität DE	24
Abb. 5.21: Kundenportal Lokalität EN	24
Abb. 5.22: Kundenportal Filter- und Suchfunktion sowie Option CSV/PDF	25

Abb. 5.23: Kundenportal Ausgabe als CSV	25
Abb. 5.24: Kundenportal Beispiel neuer Wiegebeleg	25
Abb. B.1: Mockup – Disposition	29
Abb. B.2: Mockup – Nutzer.....	30
Abb. B.3: Mockup – Nutzer anlegen.....	30
Abb. B.4: Mockup – Fahrzeuge	31
Abb. B.5: Mockup – Fahrzeuge anlegen	31
Abb. B.6: Mockup – Vertragswesen	32
Abb. B.7: Mockup – Buchhaltung	32
Abb. C.1: Kundenportal – Loginseite	33
Abb. C.2: Dashboard.....	33
Abb. C.3: Waage – Wiegung anlegen.....	34
Abb. C.4: Disposition – Container pflegen	34
Abb. C.5: Disposition – Lieferschein abrufen.....	34
Abb. C.6: Stammdaten – Nutzer anzeigen	35
Abb. C.7: Stammdaten – Nutzer erstellen.....	35
Abb. C.8: Stammdaten – Nutzer bearbeiten.....	35
Abb. C.9: Stammdaten – Kunden bearbeiten	36
Abb. C.10: Vertragswesen – Preisliste abrufen.....	36
Abb. C.11: Buchhaltung – Rechnungen abrufen/erstellen/stornieren.....	37
Abb. C.12: Controlling – Auswahl zur Statistik.....	37
Abb. C.13: Controlling – Graph exemplarisch nach Sorten.....	38
Abb. C.14: Controlling – Graph exemplarisch nach Kunden	38
Abb. C.15: Tabelle – Sorten	39

Tabellenverzeichnis

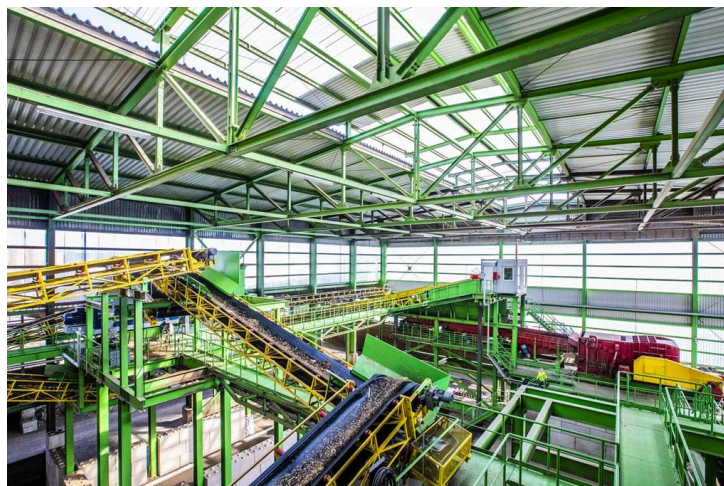
Tab. 4.1: Auswertung der Frameworks.....	9
Tab. 4.2: Fragebogen.....	12
Tab. 5.1: Plugins der IDE.....	17
Tab. A.1: Fragebogen – Annahme – Pamela Dziecol 11.04.2022.....	27
Tab. A.2: Fragebogen – Disposition – Reiner Uderhardt 12.04.2022	27
Tab. A.3: Fragebogen – Einkauf/Verkauf – Michael Ludwig 13.04.2022	28
Tab. A.4: Fragebogen – Buchhaltung – Karina Herzog 14.04.2022	28
Tab. A.5: Fragebogen – Controlling – Heiko Schurwin 14.04.2022.....	29

1 Einleitung

Die Kundenbindung ist ein bedeutsames Instrument für Betriebe, um langfristig am Markt bestehen und die eigene Position gegenüber der Konkurrenz aufrechterhalten oder gar verbessern zu können (vgl. Töpfer, 2020, S. 3). Dazu trägt entscheidend die Kundenbetreuung bei, besonders dann, wenn außerhalb der eigentlichen Öffnungszeiten Kunden Zugriffsmöglichkeiten auf Informationen und Dokumente haben (vgl. Töpfer, 2020, S. 25). Die Umsetzung eines solchen Kundenportals sowie die kontinuierliche Erweiterung und Verbesserung der darin zur Verfügung stehenden Dokumente ist ein wichtiger Ansatz, diese Kundenbindung zu erhöhen (vgl. Töpfer, 2020, S. 1 f. vgl. 2020, S. 43 f.). Diese fortlaufende Entwicklung kann über einen Development- und IT Operations-Zyklus (DevOps) erreicht werden (vgl. Alt et al., 2017, S. 27 f.).

1.1 Betriebsprofil

Der Entsorgungsfachbetrieb Holzkontor Preussen GmbH (HKP) mit Sitz in Berlin ist ein mittelständischer Betrieb, der sich auf das Recycling von Holz spezialisiert hat. Ein regelmäßig zu durchlaufender Geschäftsprozess ist die Anlieferung von Holz, das recycelt werden soll. Dieser wird wie folgt ausgeführt: Das Holz wird vom Kunden angeliefert oder von Fahrzeugen der HKP abgeholt. Nach der Anlieferung wird das Holz eingewogen, vom Fahrzeug abgeladen und geprüft. Anschließend wird das entladene Fahrzeug ausgewogen und der Wiegebeleg erstellt. Die verschiedenen Holzqualitäten werden dabei getrennt weiterverarbeitet, sortiert und geschreddert.



(Holzkontor Preussen GmbH - Anlage, 2022)

Abb. 1.1: Anlage zur Trennung/Zerkleinerung der Holzabfälle

Hochwertigere Hackschnitzel werden recycelt und z. B. für die Spanplattenherstellung verwendet. Hackschnitzel minderwertiger Qualität dienen der Energiegewinnung.



(Holzkontor Preussen GmbH - AI, 2022)

Abb. 1.2: Holz AI zur Spanplattenherstellung geeignet



(Holzkontor Preussen GmbH - AIV, 2022)

Abb. 1.3: Holz AIV nur zur Energiegewinnung geeignet

Die HKP ist hierarchisch als Linienorganisation aufgebaut und untergliedert sich in verschiedene Abteilungen. Dabei zeichnen sich die Geschäftsführer durch Innovation, Technologieaffinität und Weitsicht aus. Dies ist z. B. an dem Einsatz eines branchenweit einzigartigen und innovativen Elektro-Baggers zu erkennen.



(Holzkontor Preussen GmbH - Sennebogen, 2022)

Abb. 1.4: branchenweit innovativer Elektro-Bagger

Nachfolgend werden das Projekt und das gestellte Projektziel definiert.

1.2 Projektdefinition und Zielsetzung

Ziel dieser Arbeit ist es, für die Umsetzung eines Kundenportals einen DevOps-Zyklus aufzubauen und dafür geeignete Software zu identifizieren und auszuwählen. Durch den Einsatz eines Kundenportals soll die Kundenbindung gestärkt und die Stärke im Bereich Dienstleistungen ausgebaut werden. Durch den DevOps-Zyklus wird die kontinuierliche Entwicklung, begleitet durch parallele Tests, ermöglicht. Über das Kundenportal, das im Projektverlauf entwickelt und bereitgestellt wird, sollen die Kunden jederzeit ihre Unterlagen abrufen können. Daraus ergeben sich zwei Vorteile: Die Kunden erhalten unabhängig von den Geschäftszeiten die benötigten Informationen und Belege und der Betrieb spart Zeit und Geld, da die Belegerstellung kein Personal erfordert. Die Optimierung des Wiegeprozesses und der Erstellung der Anlieferscheine ist ein weiteres Projektziel. Im nächsten Absatz wird die Methodik zur Umsetzung beschrieben.

1.3 Methodik

Beginnend wird geeignete Software zum Aufbau eines DevOps-Zyklus herausgesucht und mithilfe von vorab definierten Kriterien verglichen. Diesbezüglich wurde eine Literaturrecherche basierend auf den zehn Schritten nach Hirt und Nordhausen durchgeführt (vgl. Nordhausen & Hirt, 2020, S. 10). Der Zeitrahmen für diese Literaturrecherche wurde bei wissenschaftlichen Artikeln auf fünf Jahre und bei themenbezogenen Fachbüchern auf fünfzehn Jahre festgelegt. Die bei der Suche verwendeten Begriffe, Schlagworte und Suchstrings sind Abwandlungen des Titels dieser Arbeit. Vorrangig dienten dabei Google Scholar, Springer Link und Webopac als Literaturlatenbanken. Für die Sammlung, Sichtung und Organisation der über die Recherche ausgewählten Literatur wurde das Literaturverzeichnisprogramm Zotero genutzt. Anhand der Literatur erfolgte die Auswahl der für den DevOps-Zyklus geeigneten Software. Im Anschluss daran begann der eigentliche Aufbau der Entwicklungsumgebung, in der das Kundenportal fortwährend entwickelt, geprüft, getestet und bereitgestellt werden kann. Das Kapitel „Entwicklungsumgebung“ beschreibt die Softwareentwicklung im Detail.

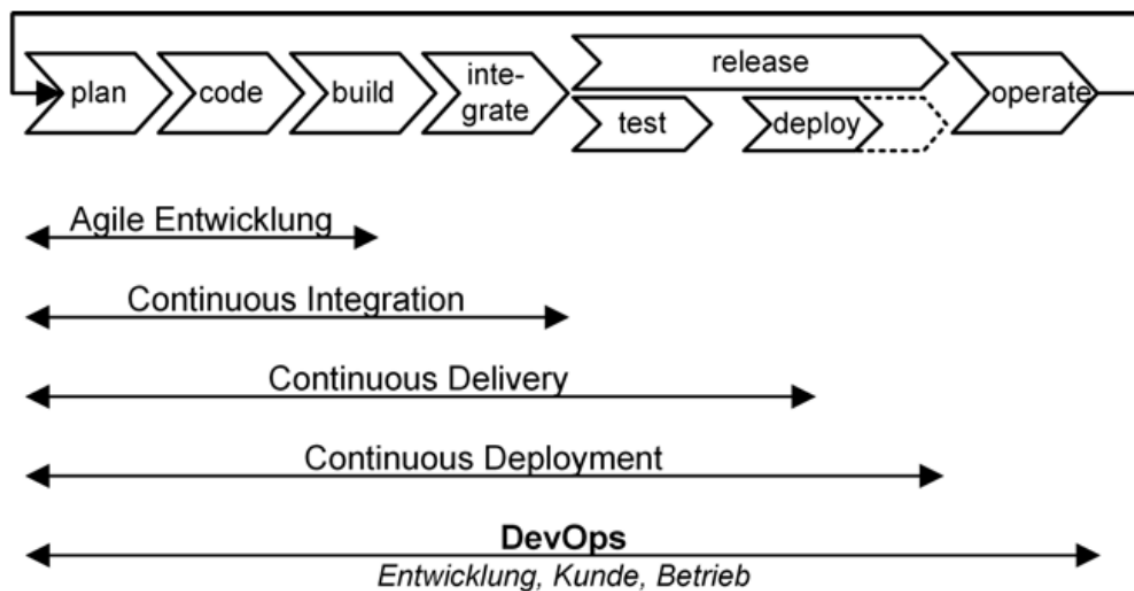
2 Grundlagen und Definition

Für das Verständnis der fachlichen Ausführungen werden zunächst die wesentlichen Begriffe erläutert. Der Fokus liegt hierbei auf DevOps sowie Enterprise Resource Planning System (ERP-System).

2.1 Development- und IT Operations-Zyklus

Development- und IT Operations kommen aus dem englischen und können mit Softwareentwicklung und dem IT-Betrieb übersetzt werden (vgl. Alt et al., 2017, S. 23). Beide Begriffe finden auch als ein Portmanteau DevOps Verwendung, wodurch die Zusammenarbeit dieser beiden Bereiche in den Vordergrund gerückt wird (vgl. Halstenberg et al., 2020, S. 1). Das Bemühen einer Zusammenarbeit über die Grenzen der eigenen Abteilung hinaus ist eines der Ziele von DevOps. Hierbei kommt es zu einer neuen Interpretation von bekannten und erfolgreichen Konzepten (vgl. Alt et al., 2017, S. 23; vgl. Halstenberg et al., 2020, S. 1).

DevOps bietet hierbei nach Halstenberg et al. die Möglichkeit, den kompletten Softwarelebenszyklus abzudecken (vgl. Halstenberg et al., 2020, S. 16).



(Quelle: Halstenberg et al., 2020, S. 16)

Abb. 2.1: Vergleich der Methoden – DevOps deckt den gesamten Lebenszyklus ab

Plan: ermöglicht es, Anforderungen mithilfe z. B. eines Kanban Boards an die zu entwickelnde Software festzulegen. Darüber hinaus ist es möglich, Probleme zu verfolgen (vgl. Halstenberg et al., 2020, S. 16 f. vgl. Katal et al., 2019, S. 4).

Code: ist die Entwicklung der Software in einer geeigneten Entwicklungsumgebung unter Verwendung eines Programms zur Versionierung (vgl. Halstenberg et al., 2020, S. 18 f. vgl. Katal et al., 2019, S. 3 f.).

Deployment Pipeline: ist der Zusammenschluss der Punkte build, integrate und test, die in der Abbildung separat aufgeführt werden. Unter build ist in dem Zusammenhang die Kompilierung der Software zu verstehen. Des Weiteren sorgt der integrate Prozess dafür, dass alle Abhängigkeiten zusammengeführt werden. Im Anschluss erfolgt automatisch ein kleiner Test der Grundfunktionalität (vgl. Halstenberg et al., 2020, S. 19; vgl. Katal et al., 2019, S. 5).

Release & Deploy: sind zwei Prozesse, die direkt aneinander anschließen können und beschreiben Bereitstellung und Freigabe einer getesteten Anwendung (vgl. Halstenberg et al., 2020, S. 20).

Operate: ist die Notwendigkeit, nach der Bereitstellung der Anwendung diese zu betreiben, da erst im laufenden Betrieb weitere Fehlerquellen erkenntlich werden (vgl. Halstenberg et al., 2020, S. 21).

Monitor: dient der Überwachung des Systems und kann zum Erfolg von DevOps beitragen. Durch eine Überwachung ist es möglich, positive und negative Auswirkungen zu erkennen und darauf zu reagieren (vgl. Katal et al., 2019, S. 4).

Feedback & Improve: ist die Möglichkeit während der Entwicklung Feedback zu erhalten und etwaige Fehler zu beseitigen (vgl. Halstenberg et al., 2020, S. 21; vgl. Katal et al., 2019, S. 7).

2.2 Enterprise Resource Planning System

ERP-Systeme sind große, modulare Anwendungen, die für die meisten Geschäftsprozesse eines Betriebs konzipiert sind (vgl. Pekša, 2018, S. 1). Dabei bilden diese für Betriebe die Grundlage und sind ein nützliches Instrument zur Koordinierung von Ressourcen, Informationen und Abläufen (vgl. Vadivelu et al., 2018, S. 1). Gemäß Schoeneberg handelt es sich bei ERP-Systemen um Softwarelösungen, in denen die unterschiedlichsten Bereiche eines Betriebes abgebildet werden können (vgl. Schoeneberg, 2011, S. 1).

3 Ausgangssituation

In diesem Kapitel wird die Ausgangssituation des Auftraggebers, der HKP, beschrieben. Der Betrieb nutzt die ERP-Lösung von rona:systems GmbH. Bei Problemen in der IT werden externe Berater oder rona:systems GmbH involviert, um diese zu lösen. Zum Projektstart gab es keinen Ansatz eines DevOps-Zyklus oder eines digitalen Kundenportals. Die Kunden mussten ihre Anfragen telefonisch, per E-Mail oder persönlich stellen. Nur die Mitarbeiter der HKP konnten die Kunden-Dokumente wie z. B. Wiegebelege, Rechnungen und sonstige Belege erstellen. Zum Projektstart waren Dokumente in ausgedruckter Form der Goldstandard. Das führte zu Wartezeiten in der Kundenbetreuung und der Kundenservice war auf die Geschäftszeiten eingeschränkt. Dieser Umstand war für den Betrieb und die Kunden ineffizient. Daraus resultierte Unzufriedenheit beim Kunden und bei den Mitarbeitern der HKP, da diese ihre Kunden nicht immer wie gewünscht betreuen konnten. Das Management der HKP erkannte die Notwendigkeit, bestehende Geschäftsprozesse durch Digitalisierung effizienter zu gestalten, um so Zeit und Kosten zu sparen und den Kundenservice zu verbessern sowie die Kundenbindung zu erhöhen. Diese Erkenntnis war ausschlaggebend für die Einführung des DevOps-Zyklus zu einem neuen Kundenportal. Dieser Auftrag wurde innerhalb des Praxisprojekts bearbeitet. Der präzisen Beschreibung des Projektauftrags und der Zielformulierung schließen sich die Ist-Analyse und die Anforderungserhebung mit der Soll-Konzeption an. In dieser Projektphase werden die Anforderungen konkretisiert. Dabei werden verschiedene Methoden, wie die Befragung der Beteiligten (Management, Mitarbeiter, Kunden etc.), die Analyse der bestehenden Prozesse, das Sichten von Dokumenten (Arbeitsanweisungen, Spezifikationen, Prozessbeschreibungen, Vorgaben zu Qualitätsstandards etc.) und die Aufnahme der Anforderungen anhand von Dokumenten und Befragungen, angewandt. Die Ist-Analyse und Anforderungserhebung startete mit den Interviews der Mitarbeiter aus den Abteilungen „Annahme“, „Disposition“, „Einkauf/Verkauf“, „Buchhaltung“, „Controlling“ und „Betriebsführung“. Dabei wurden die aktuelle Situation und die Anforderungen an das Kundenportal erhoben und dokumentiert und ein grobes Soll-Konzept erstellt. Das Soll-Konzept beschreibt die Rahmenbedingungen und Möglichkeiten des digitalen Kundenportals. Durch die Aufnahme der Anforderungen und die daraus entwickelte Soll-Konzeption ließ sich die durch DevOps gestützte Softwareentwicklung zum Kundenportal planen. Das nächste Kapitel beschreibt die detaillierte Entwicklung der Soll-Konzeption.

4 Konzepterstellung

In diesem Kapitel wird zu Beginn das grobe Konzept des DevOps-Zyklus und der Entwicklungsumgebung beschrieben. Daran schließt sich die Beschreibung zur Entwicklung des Kundenportals an.

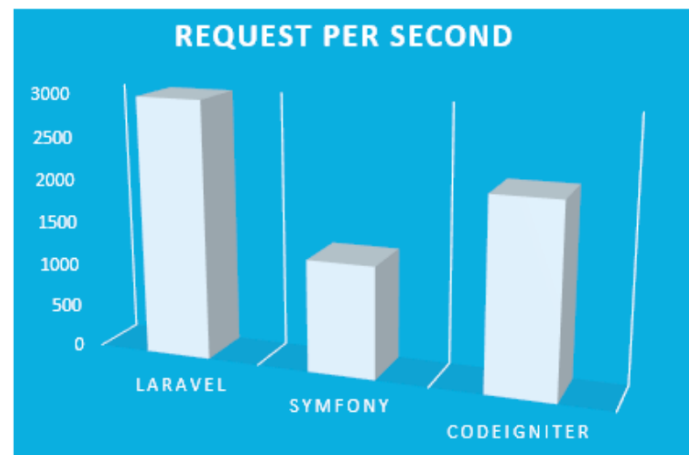
4.1 Entwicklungsumgebung

Die Entwicklung des digitalen Kundenportals erfolgte mit der integrierten Entwicklungsumgebung (IDE) PhpStorm. Folgende Sprachen wurden bei der Entwicklung genutzt:

- Structured Query Language (SQL): ist eine Sprache zur Spezifizierung von Datenbankabfragen in einer relationalen Datenbank (vgl. Dobslaw et al., 2018, S. vi).
- PHP: Hypertext Preprocessor (PHP): gilt als eine der am weitesten verbreiteten Skriptsprachen in der Web-Anwendungsentwicklung, weil sie große Flexibilität und SQL Unterstützung bietet (vgl. Laaziri et al., 2019, S. 1 f.).
- Hypertext Markup Language (HTML): ist durch die Nutzung von HTML-Tag definiert, wobei jeder Tag einen anderen Dokumentinhalt darstellt und definiert, wie der Browser den Inhalt formatieren und anzeigen muss (vgl. Eltahawey, 2016, S. 2).
- Cascading Style Sheets (CSS): bietet Gestaltungsmöglichkeiten, um z. B. Schriftart, Farbe und Größe von HTML-Elementen zu ändern, wofür es drei unterschiedliche Optionen gibt (Inline-Styles, Internal-Styles und External-Styles) (vgl. Eltahawey, 2016, S. 14).
- Javascript: ist eine Sprache, die auf Funktionalität ausgerichtet ist, aber auch zur Steuerung des Verhaltens und des Aussehens einer Website verwendet werden kann (vgl. Eltahawey, 2016, S. 1).

Nach Sichtung einiger Quellen und den dort beschriebenen Vorteilen von PHP-Frameworks wurde eines dieser zur Realisierung des Projektes implementiert. Frameworks helfen unter anderem den Entwicklern, Webanwendungen schneller und einfacher zu erstellen, indem sie ein grundlegendes Framework-Modell sowie einen vollständigen Satz von Application Programming Interface (API), Bibliotheken und Erweiterungen bereitstellen (vgl. Laaziri et al., 2019, S. 2). Die Vergleichsergebnisse von Laaziri et al., welche die drei populärsten PHP-Frameworks miteinander verglichen, dienten als Grundlage für die Bewertung der Frameworks auf ihre Geschwindigkeit (vgl. Adamu et al., 2020, S. 4).

- Anfragen pro Sekunde

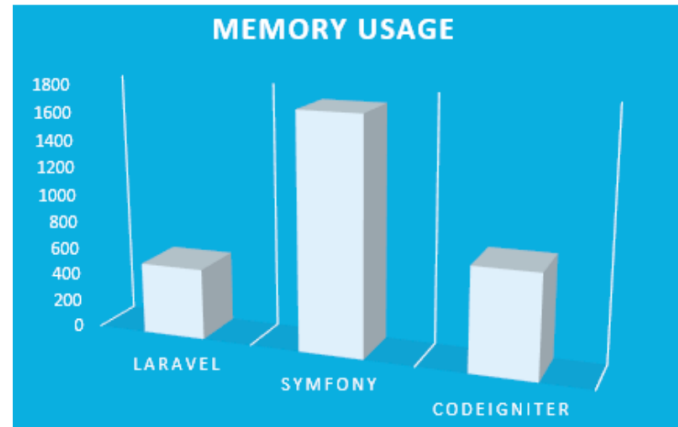


(Quelle: Laaziri et al., 2019, S. 5)

Abb. 4.1: Anfragen pro Sekunde

Das Framework Laravel verarbeitet bis zu 3000 Anfragen pro Sekunde (vgl. Laaziri et al., 2019, S. 7).

- Speicherverbrauch

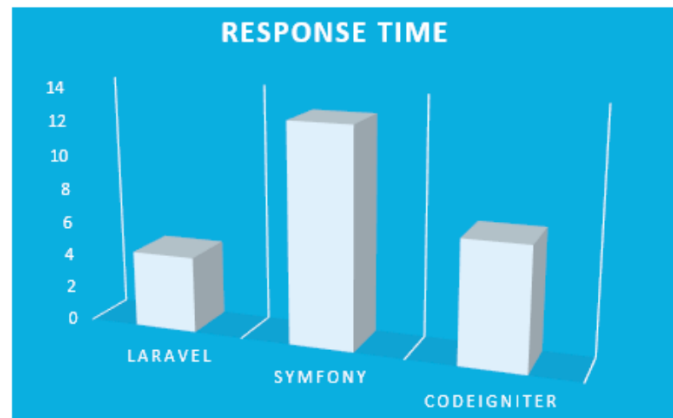


(Quelle: Laaziri et al., 2019, S. 6)

Abb. 4.2: Speicherverbrauch

In den Tests von Laaziri et al beanspruchte Laravel in der Ausführung durchschnittlich 518 Kilobyte Arbeitsspeicher. (vgl. Laaziri et al., 2019, S. 7).

- Antwortzeit



(Quelle: Laaziri et al., 2019, S. 6)

Abb. 4.3: Antwortzeit

Laravel weist mit 4,46 Millisekunden die geringste Antwortzeit auf, verglichen mit CodeIgniter mit 7,2, gefolgt von Symfony mit 12 Millisekunden (vgl. Laaziri et al., 2019, S. 7).

Des Weiteren fließen die Ergebnisse von Adamu et al. als Sicherheitsaspekte in die abschließende Bewertung mit ein, denn eine hohe Sicherheit entscheidet über die Effektivität und den Gesamterfolg jeder Anwendung und gerade Frameworks haben Sicherheitsmodule integriert, um die mit Frameworks entwickelte Anwendung so sicher wie möglich umzusetzen (vgl. Adamu et al., 2020, S. 5). Laravel bietet im Vergleich zu den anderen getesteten Frameworks den größten Umfang an Sicherheitsfunktionen, wie z. B. eine Benutzer-Authentifizierung, einen besseren Schutz vor Cross-Site-Scripting, SQL-Injektion und Cross-Site Request Forgery sowie vor weiterem Angriffspotenzial (vgl. Adamu et al., 2020, S. 6).

Tab. 4.1: Auswertung der Frameworks

Kriterium/Framework	Laravel	Symfony	CodeIgniter
Anfragen pro Sekunde	+	-	0
Speicherverbrauch	+	-	0
Antwortzeit	+	-	0
Sicherheitsaspekt	+	-	-

Unter Berücksichtigung der vorliegend betrachteten Kriterien wird zusätzlich zur IDE PhpStorm das Framework Laravel genutzt, um das Kundenportal umzusetzen. Die genaue Konzeptionierung des Kundenportals wird im Anschluss näher beleuchtet.

4.2 Development- und IT Operations-Zyklus

Die HKP ist ein kleiner Betrieb ohne eigene IT-Abteilung. Daher ist die effiziente und effektive Umsetzung der täglichen Arbeitsaufgaben ohne regelmäßiges Eingreifen in den Ablauf der IT-Prozesse eine bedeutsame Anforderung an den DevOps-Zyklus. Die Geschäftsprozesse in der HKP haben Beziehungen und Abhängigkeiten untereinander. Daraus resultieren Schnittstellen zwischen den Prozessen und die Anforderung, die Prozesse im DevOps-Zyklus mit Software zu integrieren.

Für die kollaborative und kontrollierte Entwicklung haben sich Versionsverwaltungsprogramme wie Gitlab bewährt. Programme zur Versionskontrolle ermöglichen die parallele Entwicklung durch mehrere Softwareentwickler, das Zurücksetzen auf frühere Entwicklungsstände und das Erstellen von Auswertungen (vgl. Valdivia, 2019, S. 8). GitLab bietet Optionen zur Umsetzung von Continuous Integration (CI) und weitere Funktionen, wie z. B. das Anlegen von Anforderungen und Meilensteinen sowie die Verfolgung des Fortschrittes (vgl. Than & Phyu, 2019, S. 1).

In die IDE PhpStorm lassen sich eine Reihe von zusätzlichen Entwicklungswerkzeugen, die den Entwickler in seiner Arbeit unterstützen, integrieren. Die Software TeamCity ist ein Java-basierter Server zur kontinuierlichen Entwicklung, Continuous Deployment (CD) und Integration (vgl. Cepuc et al., 2020, S. 1 f.). Dabei verwendet TeamCity intelligente Tests zum Auffinden von Duplikaten, zusätzlich gibt es Statistiken über die Build-Dauer, die Code-Qualität und die Erfolgsrate (vgl. Cepuc et al., 2020, S. 2).

Für die Realisierung des DevOps-Zyklus wurde zudem die Open-Source-Anwendung Docker genutzt. Diese steuert zentral die kontrollierte Ausführung verschiedener Programme und unterstützt die Prozessentwicklung durch das Packen von Abhängigkeiten in eine Standardform (vgl. Rad et al., 2017, S. 2). Gerade Docker dient hierbei als Bindeglied zwischen den einzelnen Anwendungen, da sowohl PhpStorm als auch TeamCity Docker unterstützen. Des Weiteren können über Docker-Container der MySQL-Server und nginx, die Webserver-Software, betrieben werden.

Für die Entwicklung des Kundenportals wurde die IDE PhpStorm um Plugins, wie Prettier (dient der Formatierung des Codes), Mintlify (für die Dokumentation des Quellcodes) und Quodana (verbessert die Code-Qualität) erweitert.

4.3 Kundenportal

Folgendes Kapitel widmet sich dem spannenden Thema der Softwareentwicklung. Jede Softwareentwicklung beginnt mit der Analyse der Ausgangssituation (Ist-Analyse) und der Spezifizierung der Anforderungen (Requirements Engineering). Aus den Ergebnissen der Anforderungsanalyse erstellt der Entwickler zusammen mit dem Auftraggeber das Soll-Konzept. Vor der Entwicklung wurden in mehreren Interviews mit den einzelnen Fachbereichen die Anforderungen aufgenommen.

In den ersten Gesprächen mit der Betriebsleitung wurden folgende genannte Kernfunktionalitäten des Portals bestimmt:

Waagemodul: bietet Kunden die Möglichkeit, die Daten zu gespeicherten Wiegeungen selbständig einzusehen und auszudrucken. Dabei darf der Kunde nur die Wiegebelege zu seiner Kundennummer aufrufen können.

Dispositionsmodul: dient der Planung von Containertransporten. Der Disponent erhält nach Auftragseingang eine Nachricht und plant die Lieferung oder Abholung der Container. Im Dispositionsmodul sind die Auftragsdaten, eine Terminplanung und die Verfügbarkeit von Ressourcen (Mitarbeiter, Container und dazu passende Fahrzeuge) zu integrieren.

Vertragsmodul: soll eine dauerhafte Verfügbarkeit der Preise und Verbindung zum Ein-/Verkauf, um diese nachzuverhandeln, gewährleisten. Zusätzlich kann ein Ausdruck als Portable Document Format (PDF) erfolgen.

Buchhaltungsmodul: angelehnt an die Möglichkeiten des Waagemoduls, kann der Kunde Rechnungen heraussuchen, um diese als PDF auszudrucken.

Controllingmodul: ermöglicht unter anderem die Auswertung zu Anliefermengen, Anlieferqualität und Preisen.

Stammdatenpflege: soll Kunden die Option zur Verfügung stellen, eigenständig Nutzer anlegen und entsprechende Rechte an Angestellte vergeben zu können. Fahrzeuge müssen hinterlegt werden, um einen Lieferschein anlegen zu können, über den eine spätere Anlieferung erfolgen kann.

Der Spezifizierung der Kernfunktionalitäten, die im Kundenportal realisiert werden müssen, folgten die Interviews in den Fachabteilungen. Darin wurden die speziellen Anforderungen der Fachbereiche aufgenommen. In den Interviews wurde folgender auf die besonderen Anforderungen der Fachbereiche abgestimmter Fragebogen verwendet.

Tab. 4.2: Fragebogen

Frage	Antwort
1. Welche Grundfunktionalitäten müssen angeboten werden?	
2. Welche Daten dürfen Kunden sehen?	
3. Nach welchen Kriterien soll gesucht und gefiltert werden?	
4. Welche Rechte werden benötigt, um Zugriff auf diesen Bereich zu haben?	
5. Welche Unterseiten sind notwendig?	
6. Wie soll ein PDF, falls erforderlich, aufgebaut werden?	
7. Welche Informationen sollen dort abgebildet werden?	
8. Ist es erforderlich, Daten als Comma-separated values (CSV) Datei ausdrucken zu können?	
9. Welche Daten sollen hierbei ausgegeben werden?	

Im nächsten Schritt werden die Fragebögen ausgewertet. Die genauen Antworten können im Anhang (A) eingesehen werden.

Annahme: dient der zusätzlichen Bearbeitung von Wiegebelegen. Zum Benutzerkreis, der das Recht zur Bearbeitung der Wiegebelege erhält, gehören Wäger, das Rechnungswesen (Financing) und die Betriebsleitung (Management). Die Anwendung muss auch für ungeübte Benutzer intuitiv sein. Eine unkomplizierte und intuitive Suche nach bestimmten Daten (wie z. B. Wiegebelege) und das Filtern und die Sortierung dieser sind zu gewährleisten. Das soll die selbständige Nutzung des Kundenportals durch den Anwender ohne Rückfragen bei der Administration sicherstellen. Unterseiten strukturieren die Prozesskette zur Suche, Bearbeitung und zum Ausdruck von Wiegebelegen. Der Anwender soll CSV-Daten exportieren und in ein Tabellenkalkulationsprogramm weiterverarbeiten können. Der Wiegebeleg sollte sich an dem aktuellen orientieren.

Holzkontor Preussen GmbH

3585191 /E


Kunde: 314691
 Wurzer Umwelt GmbH
 Entsorgung und Wiederverwertung
 Am Kompostwerk 1
 85462 Eitting

Erstwiegung: 20.04.22 07:29
 Zweitwiegung: 20.04.22 08:03


Baustelle: Altholzaufbereitung E... W086 86582 Brutto: W 36.360 kg
 Am Kompostwerk 1 W9971 16 Tara: W 16.400 kg
 85462 Eitting Netto: E 19.960 kg

Spediteur: KFZ: LL WT 5009


Art.Nr.	Bezeichnung	AVV	Menge	Gewicht
241910	Hackschnitzel thermisch 17. Binschw	19 12 06*		19.960 kg
940005	Gebühren BS ZEDAL pro St		1,00 Stüc	0 kg



Sophtenwerdweg 60
 D-13287 Berlin
 Telefon 030-33002970
 Telefax 030-33002977
 E-Mail: info@hk-preussen.de
 Homepage: http://www.hk-preussen.de



Unterschrift Anlieferer



Unterschrift Wäger

Reservierte aus drei programmierbarer Zusatzanordnung, die gleichen Messwerte können eingesehen werden.

Abb. 4.4: aktueller Wiegebeleg

Disposition: muss die Containeraktionen (Stellen, Tauschen, Abholen) in Einklang mit der tatsächlichen Behälteranzahl bringen, wenn keine Container zur Verfügung stehen, kann auch kein neuer Container gestellt werden. Die Logik muss hierbei eine entscheidende Rolle spielen. Informationen über Anmeldungen dieser Tauschaktionen sollen bestmöglich direkt per E-Mail erfolgen, damit diese sinnvoll disponiert werden können. Lieferscheine sollen im Anschluss digital ausgedruckt werden. Ein Kalender mit geplanten Containeraktionen soll genauso umgesetzt werden, wie die Aufträge per Drag-and-drop den einzelnen Fahrzeugen zuordnen zu können.


Vertragswesen: muss gewährleisten, dass Kunden nur ihre jeweiligen Preise sehen, die als PDF ausgedruckt werden können. Nachverhandlungen sollen erst ab einer gewissen Zeit möglich sein, direkt über einen Button, wo der Einkäufer eine Information per E-Mail erhält und sich dann mit dem Kunden in Verbindung setzen kann. Es müssen Preise den Kunden und Baustellen zugeordnet werden können, da einige Kunden unterschiedliche Preise für unterschiedliche Baustellen haben. Hierbei spielt die Qualität eine erhebliche Rolle. Einbringen von Reklamationen und bildlicher Dokumentation, was bei der Annahme wichtig werden könnte. Zusätzlich wäre interessant, wenn Kunden bei Reklamation automatisch eine Notiz erhalten, um entsprechend reagieren zu können.

Buchhaltung: soll Kunden die Möglichkeit geben, die jeweiligen Rechnungen suchen und als PDF abspeichern zu können. Es muss bedacht werden, dass nicht jeder Angestellte Zugriff auf die Rechnungen haben darf. Bei der Vorlage soll die aktuelle Rechnung als Grundlage genutzt werden.

Holzkontor Preussen GmbH

Holzkontor Preussen GmbH Sophienwerderweg 60 13597 Berlin

DB Netz AG
Regionalbereich Ost c/o Deutsche Bahn AG
DB SSC Buchhaltung Deutschland
Elisabeth-Schwarzhaupt-Platz 1
10115 Berlin



Sophienwerderweg 60
D-13597 Berlin
Telefon 030-33022970
Telefax 030-33022977
E-Mail: info@hkp-preussen.de
Homepage: http://www.hkp-preussen.de

RECHNUNG

Exemplar für Ablage

Kundenadresse: DB Netz Zentrale
Adam-Riese-Straße 11-13
D 60327 Frankfurt am Main

Bestellung 0016/HP1/11194490, Projektnr. 22210170 AFM 34974, BV: BF Cottbus Altschwellen Holz

Bei Zahlung bitte angeben:

Kunden-Nr. 314435/Debitoren-Nr.: 1314435

Rechnung Nr. 20221751

Sachbearbeiter Neumann Jens

Datum: 12.05.2022

Buchungsdatum: 30.04.2022

Seite 1

Artikel-Nr.	Bezeichnung	Abfallschlüssel	Menge	Einheit	Einzelpreis	Summe EUR
Lieferungen von 27.04.22 bis 27.04.22						
201524	Holzschwellenbruch	17 02 04*	11,620	t	80,00	Erlöskonto: 9999 Kst.: 2200 929,60
940007	Verladung & Transport		1,000	St.	439,60	Erlöskonto: 44150 Kst.: 2200 439,60

MWST 10,0% 7,0% 0,0%

Basis 1.399,10

Steuer 260,13

Zahlbar 7 Tage netto

Nettobetrag: EUR 1.399,10

+ MWST EUR 260,13

Rechnungsbetrag EUR 1.629,23

Fälligkeitsdatum: 19.05.22

Geschäftsführer: Michael Holz
Regina Wegner

Ambergottel Berlin HRB 85721
Ust-ID-Nr.: DE 230095892
Steuer-Nr.: 29/552/30240

Commerzbank AG
IBAN: DE56 1054 0000 0007 762600
BIC: COBADE33XXX






Abb. 4.5: aktuelle Rechnung

Controlling: soll aufgrund des komplexen Abfallrechts und behördlicher Anordnung, Dokumente und Statistiken zu festgelegten Zeitpunkten einreichen zu müssen, die Möglichkeit bieten, Auswertungen und Statistiken zu ziehen. Ziel ist es dabei, die vorhandenen Daten abgabefertig vorzubereiten, um jederzeit den Status des Entsorgungsfachbetriebes aufrechterhalten zu können.

Aus den in den Interviews erhobenen Informationen konnte das Soll-Konzept zur Realisierung der Anforderungen und das Grobkonzept zur Gestaltung der Oberfläche bestimmt werden. In weiteren Feedbackgesprächen ist die Konzeption verfeinert worden. Nachdem die erste Runde der Interviews für Informationen gesorgt hat, wie die Seiten explizit gestaltet werden müssen, um allen Anforderungen gerecht zu werden, fanden regelmäßige Feedbackgespräche statt.

Der Analyse- und Definitionsphase schließt sich die Entwurfsphase an. Diese geht der eigentlichen Implementierung voraus. In der Entwurfsphase werden die zu entwickelnden

Systemkomponenten bestimmt, die Struktur der Daten und die Beziehungen der Daten untereinander modelliert sowie die Gestaltung der Benutzeroberfläche (GUI) entworfen. Für den Entwurf der GUI hat sich die Erstellung von Mockups bewährt. Mockups sind eine grafische Abbildung der künftigen GUI ohne Funktionalität. Sie dienen vorrangig der Visualisierung und ermöglichen so die Vorstellung vom Programm, ohne dass eine Codezeile geschrieben wurde. Folgende Abbildung zeigt beispielhaft ein Mockup aus dem Projekt. Weitere Mockups sind zusätzlich zur Verfügbarkeit im GitLab auch im Anhang (B) zu finden.

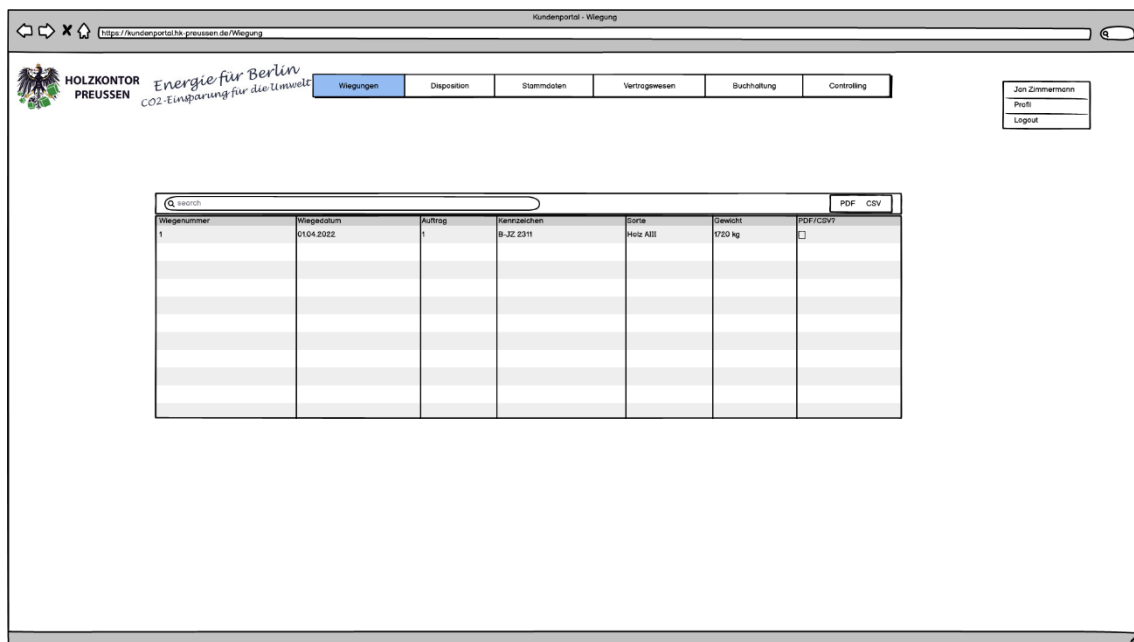


Abb. 4.6: Mockup – Wiegungen

Wie aus den Anforderungen aus dem Bereich Waage hervorging, sollten die Wiegebelege bestmöglich über eine Suchfunktion sortier- und auffindbar sein. Des Weiteren muss die Möglichkeit bestehen, diese über eine Combobox zu selektieren und als PDF abspeichern und drucken zu können. Zusätzlich zur Druckfunktion als PDF soll, wenn möglich, eine Option eingebettet werden, mit der die Wiegebelege in Form einer CSV-Tabelle heruntergeladen und im Anschluss vom Kunden geprüft und für weitere Auswertungen genutzt werden können.

Da nun alle wesentlichen internen und externen Bedingungen innerhalb dieses Konzeptes beschrieben sind sowie ein grober Plan zur Gestaltung existiert, stellt sich die Frage – wie lässt sich dieses Konzept nun erfolgreich und pragmatisch umsetzen? Mit dieser Frage befasst sich das folgende Kapitel.

5 Umsetzung

Dieses Kapitel beschreibt die Implementierung der Software mithilfe der IDE PhpStorm und des Frameworks Laravel. Das Kapitel beginnt mit einem Einblick in die IDE, dem sich die Demonstration des DevOps-Zyklus anschließt. Am Ende des Kapitels wird die Implementierung der Softwarekomponenten beschrieben.

5.1 Entwicklungsumgebung

Vor Beginn der eigentlichen Entwicklung war die Entwicklungsumgebung mit allen dafür benötigten Komponenten zu installieren und zu konfigurieren. Für die Entwicklung des Kundenportals wurden die IDE PhpStorm, das PHP-Framework Laravel, das Datenbankmanagementsystem (DBMS) MySQL und einige für PhpStorm verfügbare Plugins installiert. Die Einrichtung der Entwicklungsumgebung beginnt mit der Installation und Konfiguration der IDE. PhpStorm lässt sich durch Installationsroutinen und fertige Installations- und Konfigurationsskripts problemlos und schnell installieren und einrichten. Bestehende Abhängigkeiten zwischen einzelnen Modulen werden durch die automatisierte Installation erkannt und aufgelöst. Im zweiten Schritt wurde das DBMS MySQL eingerichtet. Zusätzlich sind 87 Plugins installiert. Die Tabelle Tab. 5.1 enthält die Funktionsbeschreibungen zu zehn Plugins.

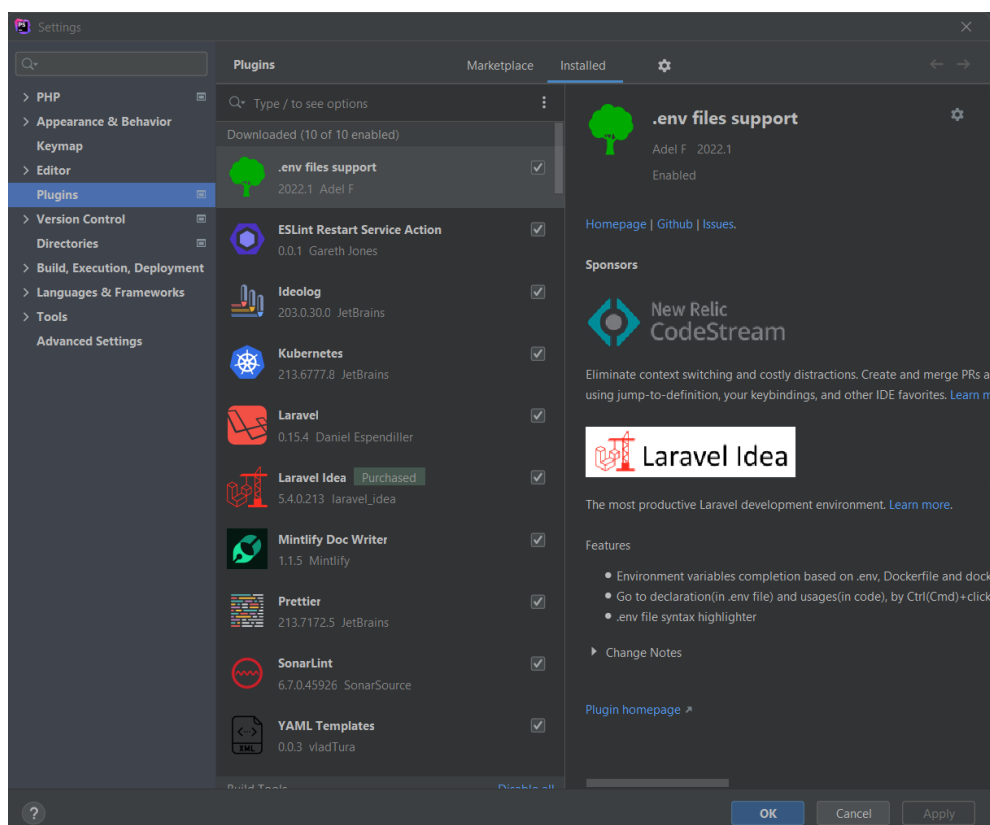


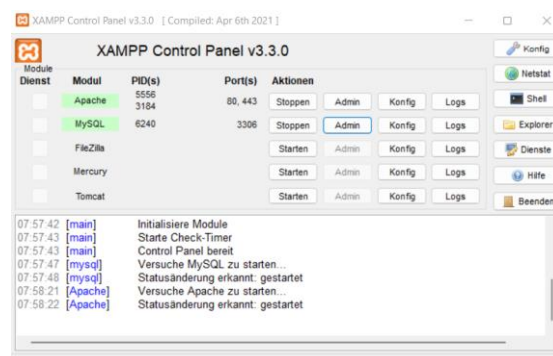
Abb. 5.1: Screenshot der Plugins (Strg+Alt+S)

Tab. 5.1: Plugins der IDE

Plugin	Funktion
Git & GitHub	Bietet typische Git Standardfunktionen, um ein Repository zu klonen oder zu erstellen. Des Weiteren können push, pull, merge, commit Anfragen gestellt werden.
Laravel	Umsetzung aller für Laravel typischen Datei- und Komponentenvarianten wie z. B. Route, Views, Controller, Blade, Templates, ...
Laravel Idea	Erweitert die Funktionen des Laravel Plugins um Codegenerierung und unterstützt dabei das genutzte Livewire Paket.
.env files support	In der env-Datei wird unter anderem der Zugang zur Datenbank gespeichert. Hierfür bietet dieses Plugin eine Unterstützung zusätzlich für Laravel.
SonarLint	Ein mächtiges Plugin, das Bugs und Schwachstellen im Code findet und behebt. Des Weiteren werden Korrekturen und Rechtschreibprüfungen während des Schreibvorgangs angezeigt.
Grazie	Intelligente Rechtschreib- und Grammatikprüfung für das Geschriebene, funktioniert auch für Programmiersprachen.
Mintlify Doc Writer	Es kann nachträglich eine automatische Dokumentation für den Code umgesetzt werden.
Prettier	Nachdem der Code geschrieben wurde, bietet dieses Plugin die Möglichkeit, den Code ansehnlich zu formatieren.
Docker	Direkte Option, aus der IDE heraus, Docker-Container zu starten.
Quodana	Ist eine Plattform zur Überwachung der Codequalität im laufenden Entwicklungsprozess.

5.1.1 Webserver

Anders als im Konzept beschrieben wurde zunächst xampp genutzt, um den MySQL-Server und phpMyAdmin zu betreiben.

**Abb. 5.2:** Screenshot XAMPP

PhpMyAdmin bot dabei die Möglichkeit, die notwendige Datenbank „hkp“ anzulegen.

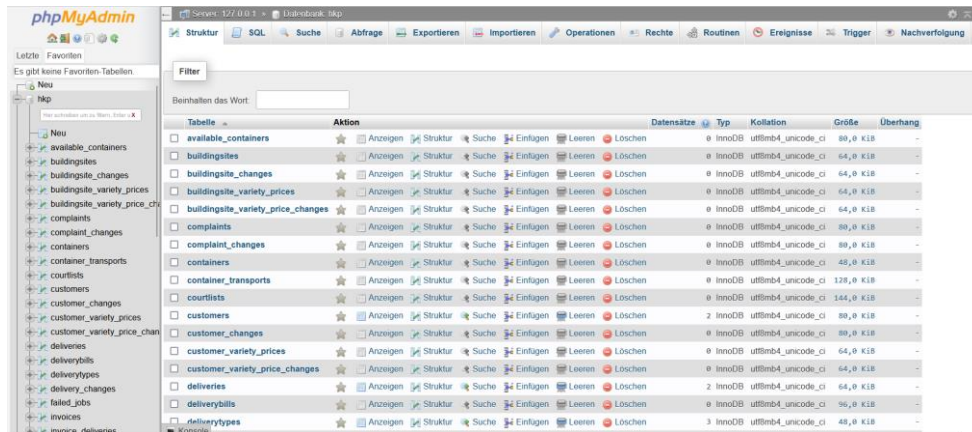


Abb. 5.3: Screenshot phpMyAdmin

Direkt in PhpStorm wird ein Terminal zur Verfügung gestellt, über das mit dem Befehl „php artisan serve“ ein Laravel-Entwicklungsserver gestartet werden kann.

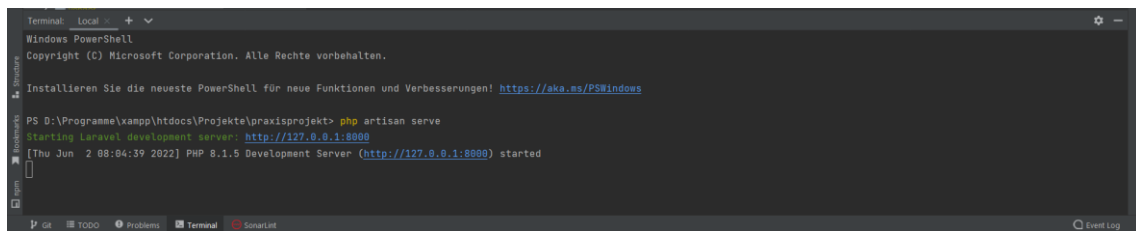


Abb. 5.4: Terminal PhpStorm

5.1.2 integrierte Entwicklungsumgebung

Die IDE ist komplex und deren Funktionen werden nachfolgend anhand von Screenshots und einer kurzen Beschreibung zur Nutzung vorgestellt.

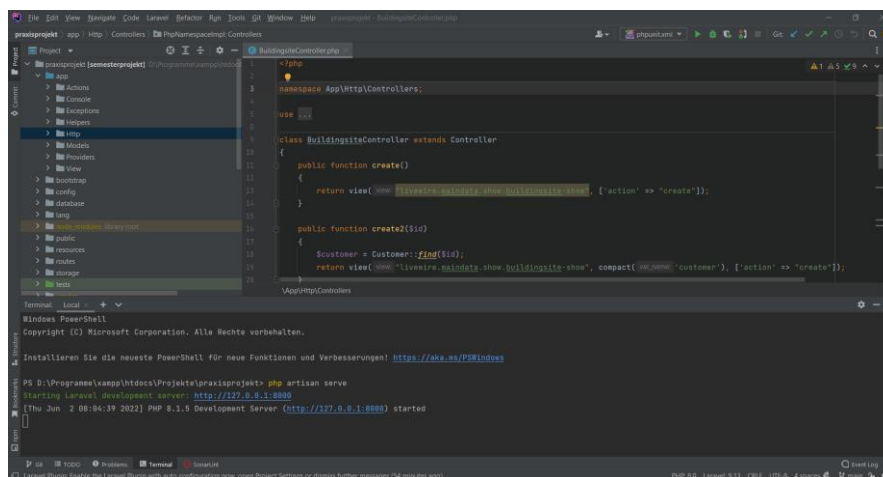


Abb. 5.5: IDE komplett

Grob kann PhpStorm in drei Arbeitsbereiche unterteilt werden, wobei dies individuell angepasst werden kann. Der linke Bereich enthält die Projektübersicht, die Auflistung aller Dateien und Komponenten des Projektes.

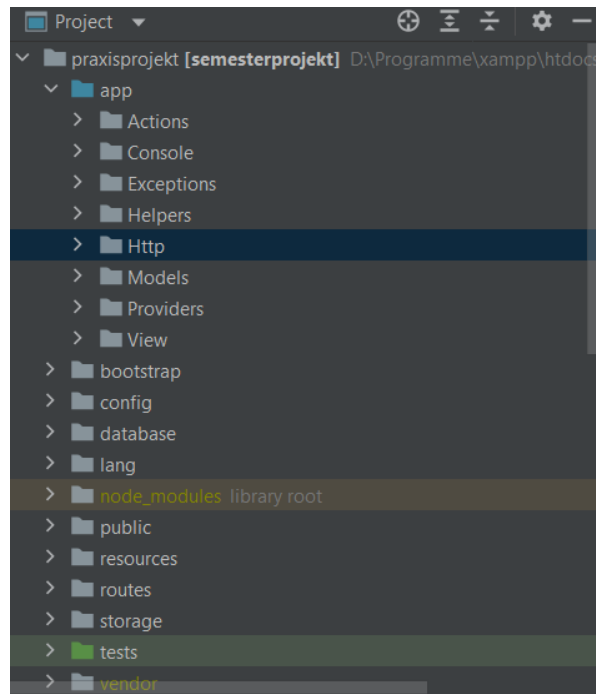


Abb. 5.6: IDE – Projektübersicht

Im rechten Abschnitt befinden sich die geöffneten Dateien, die bearbeitet werden können.

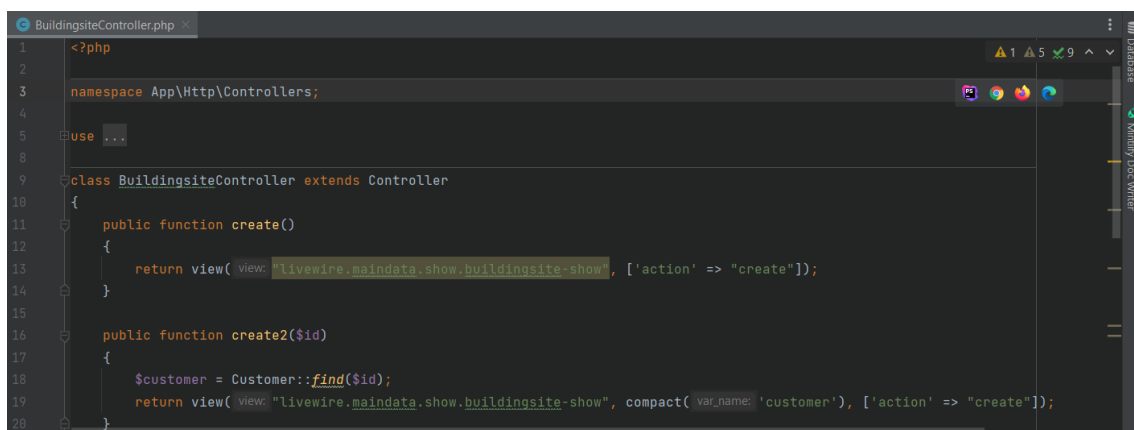


Abb. 5.7: IDE – geöffnete Datei

Über die Hervorhebung (Highlighting) von Codeabschnitten wird die Verknüpfung zu anderen Komponenten visualisiert.

Zusätzlich gibt es direkt Möglichkeiten, um auf die Code-Qualität, die Fehler und auf das Repository zugreifen zu können.

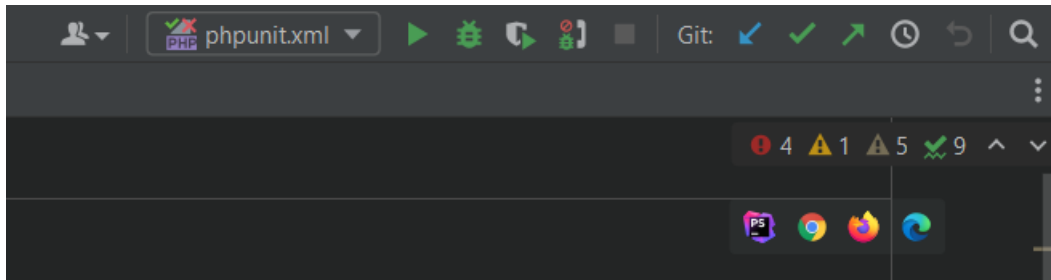


Abb. 5.8: IDE – geöffnete Datei

Im unteren Bereich von PhpStorm bietet sich die Möglichkeit, zusätzliche Informationen abzurufen.

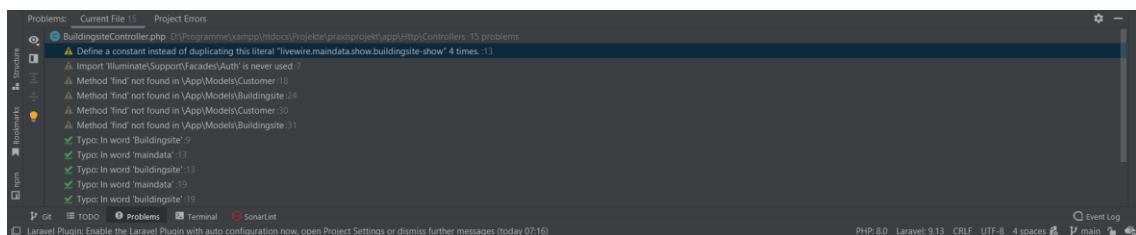


Abb. 5.9: IDE – gefundene Fehler und Probleme

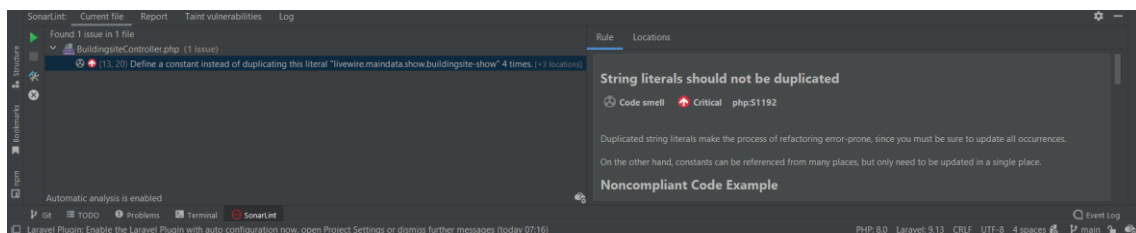


Abb. 5.10: IDE – SonarLint

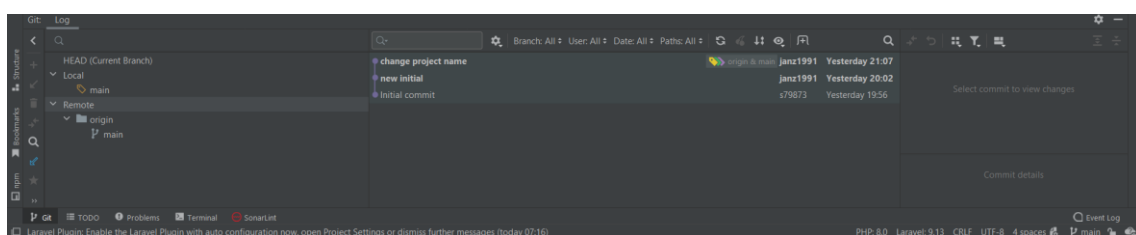


Abb. 5.11: IDE – Git Verknüpfung

Zu diesen zusätzlichen Informationen zählen das Anzeigen von Fehlern und die leichte Behebung dieser. Zudem bietet SonarLint die Funktion, die Code-Qualität zu steigern. Des Weiteren wird die Option zum Git-Repository gegeben.

Auf die großen Vorteile der genutzten Frameworks Laravel, Livewire und Jetstream wird erst in der Entwicklung des Kundenportals eingegangen.

5.2 Development- und IT Operations-Zyklus

Der DevOps-Zyklus orientiert sich stark an dem Vorsatz, Anwendungen im laufenden Betrieb schnellstmöglich ändern zu können und dabei auf ältere Versionen zurückspringen zu können. Wobei der Entwickler jederzeit Informationen zur Qualität des Codes erhält. Dadurch soll sich die Code-Qualität stetig verbessern.

Zur Versionskontrolle kam die Software GitLab zum Einsatz. GitLab ermöglicht über die Versionskontrolle hinweg die Option, Anforderungen und Meilensteine nachzuverfolgen.

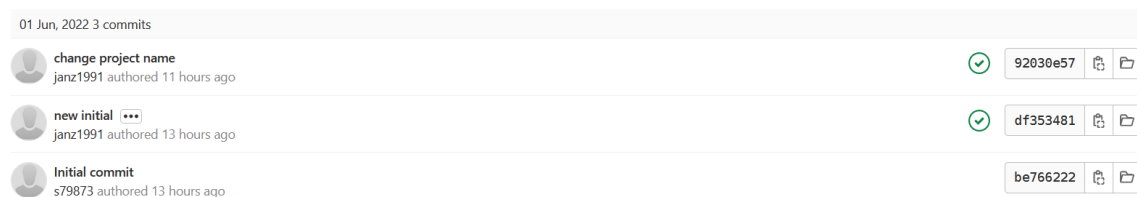


Abb. 5.12: GitLab – Versionen

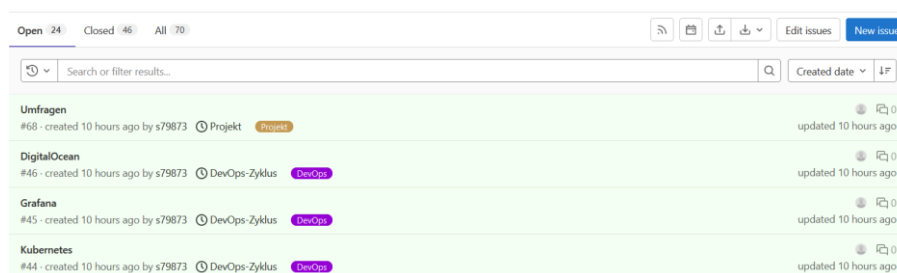


Abb. 5.13: GitLab – Anforderungen

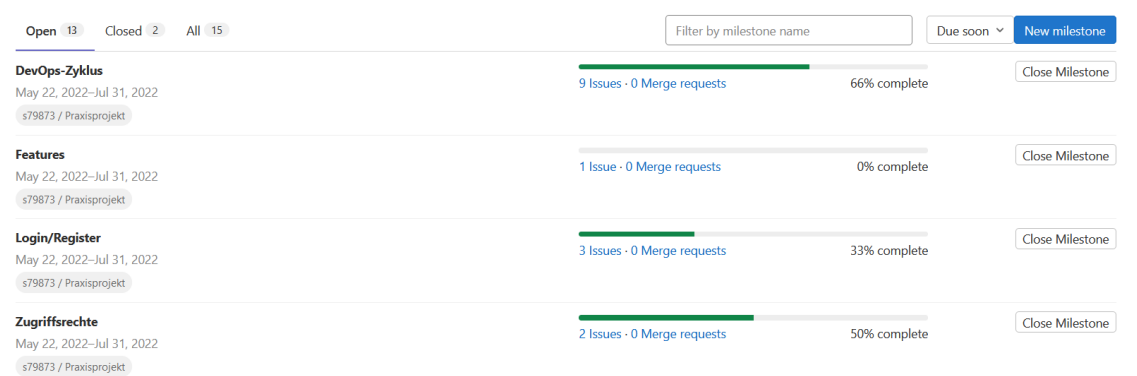


Abb. 5.14: GitLab – Meilensteine

Bei jeder Änderung, die in GitLab gepusht wird, findet über die CI/CD-Pipeline eine Prüfung des Codes statt. Das aktuelle Projekt wird dabei aufgebaut, getestet und testweise bereitgestellt.

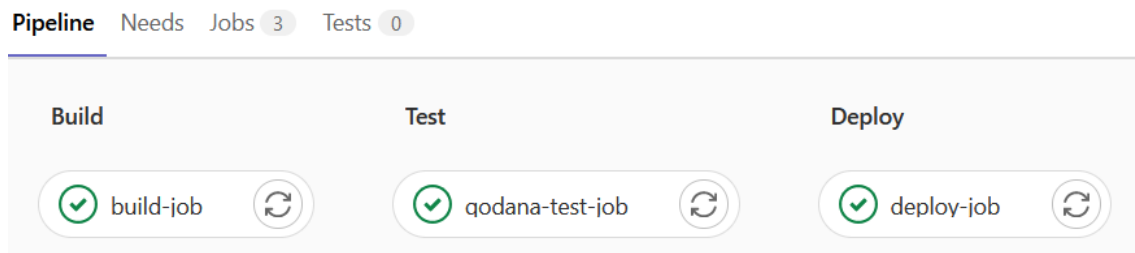


Abb. 5.15: GitLab – CI/CD-Pipeline

Docker sorgt für eine Verknüpfung des Projektes über GitLab hinweg. So werden Docker-Container genutzt, um bei jeder Änderung in GitLab zusätzliche Prüfungen der Qualität des Codes anzustoßen und diese zu überwachen.

The screenshot shows the Docker Desktop interface. The left sidebar contains navigation options: Home, Containers, Images, Volumes, and Dev Environments. The main area displays 'Images on disk' with a summary of 8 images and a total size of 5.45 GB. Below this, there is a table of local images.

NAME	TAG	IMAGE ID	CREATED	SIZE
grafana/agent	latest	0685e68273e0	30 days ago	202.63 MB
grafana/grafana	latest	21d6214505a0	1 day ago	292.93 MB
jetbrains/teamcity-server	latest	ff99894aec54	about 1 month ago	2.54 GB
nginx	latest	0e901e68141f	5 days ago	141.53 MB
node	lts	9d200cd667d5	5 days ago	907 MB
phpmyadmin	latest	593a7967c213	4 days ago	510.2 MB
phpstorm/mysql	latest	83bdf19c3a3e	about 1 year ago	405.85 MB
sonarqube	latest	4ca41017fd9b	about 2 months ago	531.64 MB

Abb. 5.16: Docker-Container

Dabei realisiert der TeamCity-Server eine zusätzliche Code-Überprüfung. Ein Agent übernimmt in Docker den Aufbau der Anwendung. Die Prüfung des Codes erfolgt auf dem Server. TeamCity verfügt über weitere Funktionen, die zur Auswertung genutzt werden können. Hier kann unter anderem abgelesen werden, wie lange die Prüfung dauert und ob diese erfolgreich war.

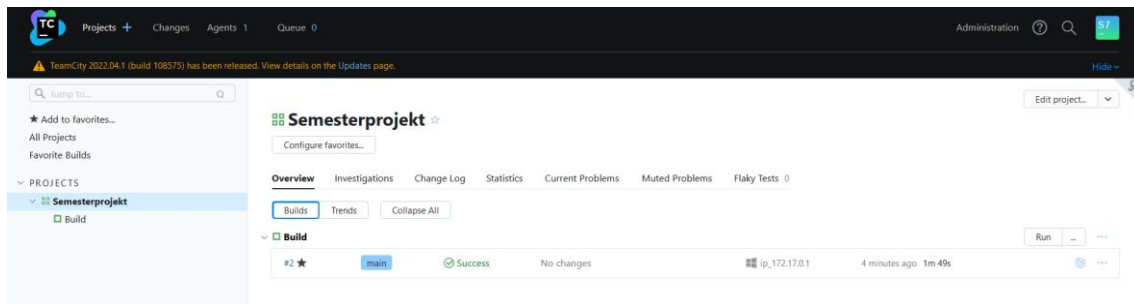


Abb. 5.17: TeamCity CI/CD-Pipeline

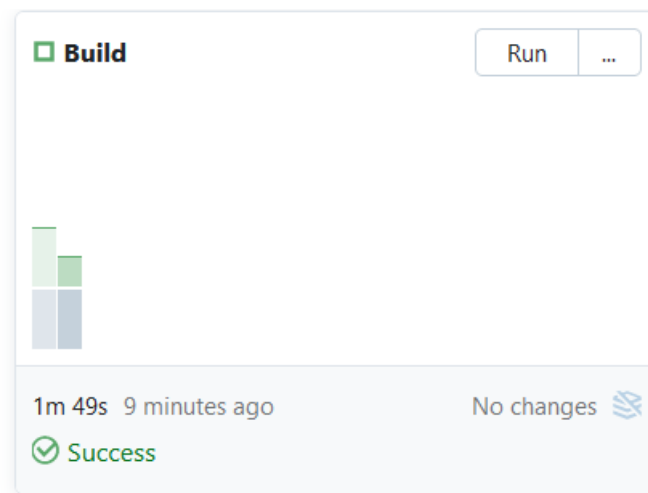


Abb. 5.18: TeamCity Auswertung

Grafana als weiterer Docker-Container überwacht die gesamten Prozesse. Hierüber können diverse Überwachungen durchgeführt werden. Wovon bisher erst die Warnungen implementiert sind.

Status	ID	Title	Alerts	Source	Created	Users
Resolved	#3	Title undefined (Check Web Title Template)	2	Formatted Webhook	Juni 02, 2022 09:31 AM	s79873

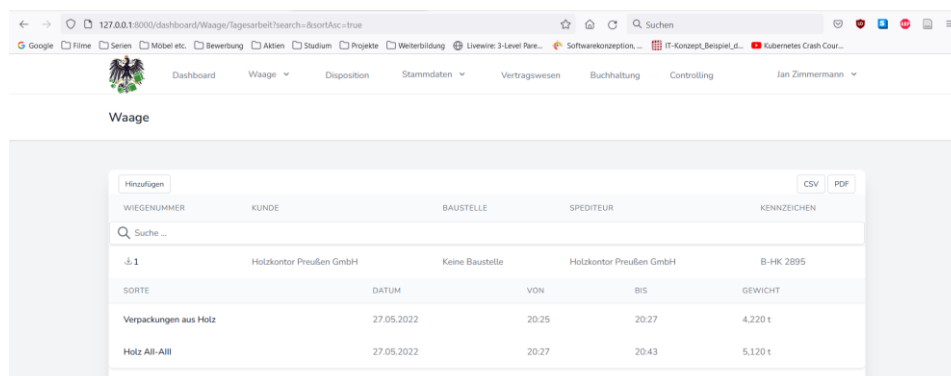
Abb. 5.19: Grafana Warnungen

5.3 Entwicklung

Nachdem wie Eingangs beschrieben ein näherer Blick in die Entwicklungsumgebung und den DevOps-Zyklus geworfen wurde, folgt nun die eigentliche Programmierung des Kundenportals.

Laravel, Livewire und Jetstream bringen über die zusätzlichen Komponenten wie z. B. die Blade-Dateien, die Route oder die Migrations Dynamik in die Programmierung. Gerade die Migrations, Seeder und Trigger sorgen dafür, dass die Datenbank über das Repository hinaus jederzeit in einem neuen System implementiert werden kann. Jetstream

bringt über die x-jet-Komponenten eine hohe Replizierbarkeit für einzelne Codeabschnitte mit. Dadurch ist es möglich, schnell eine Navigation aufzubauen, die verschiedene Gestaltungsoptionen ermöglicht. So können z. B. gewählte Navigationspunkte hervorgehoben oder Informationen angezeigt werden, wenn der Mauszeiger über einen Menüpunkt fährt. Livewire ist hervorragend geeignet, um Daten aus der Datenbank zu lesen oder diese zurückzuschreiben, da dies mitunter in Echtzeit passieren kann, was für Such- oder Sortierfunktionen hilfreich ist. Dafür implementiert Livewire eine Verknüpfung von Komponenten zu Blade-Dateien, die nicht an die Route gebunden sind. Die schnelle Anpassung von Sprachen erleichtert die Administration und die Anwendung des Kundenportals gleichermaßen. In der Konfiguration können Standardsprachen und dazugehörige Regionen (Locations) konfiguriert werden. Anhand der IP-Adresse, die der Benutzer beim Aufruf des Kundenportals übermittelt, werden die Sprache der Anwendung und in der Adresszeile automatisch auf die zur IP-Adresse passenden Sprache umgestellt.

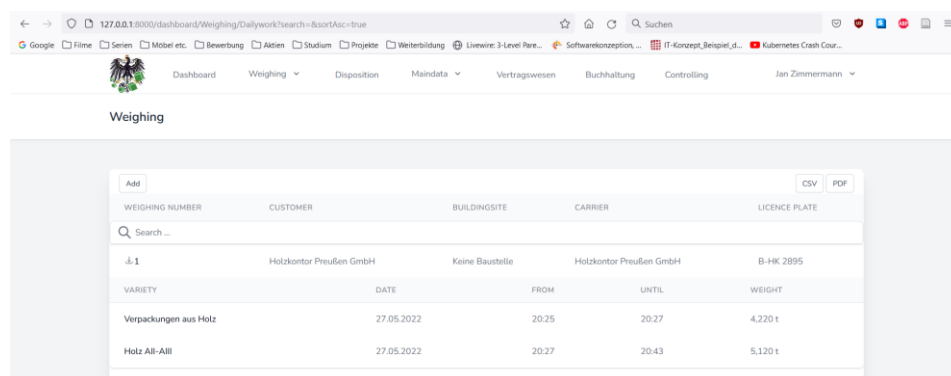


The screenshot shows the 'Waage' page with a search bar and a table of weighing data. The table has columns for 'WIEGENUMMER', 'KUNDE', 'BAUSTELLE', 'SPEDITEUR', and 'KENNZEICHEN'. A search result is shown for 'Holzkontor Preußen GmbH' with a weighing number of 1. Below the table, there is a section for 'Verpackungen aus Holz' with columns for 'SORT', 'DATUM', 'VON', 'BIS', and 'GEWICHT'.

WIEGENUMMER	KUNDE	BAUSTELLE	SPEDITEUR	KENNZEICHEN
1	Holzkontor Preußen GmbH	Keine Baustelle	Holzkontor Preußen GmbH	B-HK 2895

SORT	DATUM	VON	BIS	GEWICHT
Verpackungen aus Holz	27.05.2022	20:25	20:27	4,220 t
Holz All-All	27.05.2022	20:27	20:43	5,120 t

Abb. 5.20: Kundenportal Lokalität DE



The screenshot shows the 'Weighing' page with a search bar and a table of weighing data. The table has columns for 'WEIGHING NUMBER', 'CUSTOMER', 'BUILDINGSITE', 'CARRIER', and 'LICENCE PLATE'. A search result is shown for 'Holzkontor Preußen GmbH' with a weighing number of 1. Below the table, there is a section for 'Verpackungen aus Holz' with columns for 'VARIETY', 'DATE', 'FROM', 'UNTIL', and 'WEIGHT'.

WEIGHING NUMBER	CUSTOMER	BUILDINGSITE	CARRIER	LICENCE PLATE
1	Holzkontor Preußen GmbH	Keine Baustelle	Holzkontor Preußen GmbH	B-HK 2895

VARIETY	DATE	FROM	UNTIL	WEIGHT
Verpackungen aus Holz	27.05.2022	20:25	20:27	4,220 t
Holz All-All	27.05.2022	20:27	20:43	5,120 t

Abb. 5.21: Kundenportal Lokalität EN

In nahezu allen Bereichen der Webseite bietet sich die Möglichkeit, Daten als CSV zu drucken, dabei kann über die Such- und Filterfunktion das Ergebnis angepasst werden. Zusätzlich dazu wird Pagination genutzt, um die Ergebnisse auf der Seite auf zehn zu begrenzen.

LIEFEROPTIONSNUMMER	LIEFEROPTION	ERSTELLT VON	ERSTELLT AM
1	Direktbeladung	Jan Zimmermann	27.05.2022
2	Liefern	Jan Zimmermann	27.05.2022
3	Strecke	Jan Zimmermann	27.05.2022

Abb. 5.22: Kundenportal Filter- und Suchfunktion sowie Option CSV/PDF

A	B	C	D
1	id name	created_at	created_by
2	1 Direktbeladung	2022-05-27T19:02:13.000000Z	Jan Zimmermann
3	2 Liefern	2022-05-27T19:02:13.000000Z	Jan Zimmermann
4	3 Strecke	2022-05-27T19:02:13.000000Z	Jan Zimmermann

Abb. 5.23: Kundenportal Ausgabe als CSV

Dabei ist der Seitenaufbau der einzelnen Module nahezu identisch gestaltet. Es sind Such- und Sortierfunktionen implementiert. Im rechten oberen Rand befindet sich die Möglichkeit, alle zu dem Zeitpunkt gefilterten Daten als CSV auszugeben, die PDF-Funktion ist derzeit noch nicht implementiert in diesem Zusammenhang. Links oberhalb der Tabelle können neue Eingaben getätigt werden und mit Klick auf die Identifikationsnummer (ID) innerhalb der Tabelle, kann dieser Eintrag verändert werden. Links neben der Zahl gibt es die Möglichkeit, diesen Eintrag als PDF herunterzuladen. Eine nähere Auflistung der Webseite sowie einige Codebeispiele befinden sich im Anhang (C).

Holzkontor Preussen GmbH
Sophienwerderweg 60
13597 Berlin

Holzkontor Preussen GmbH | Sophienwerderweg 60 | 13597 Berlin

Kunde: 1
Holzkontor Preußen GmbH
Sophienwerderweg 60
13597 Berlin

Wiesegschein: 1

Erstwiegung: 27.05.2022 20:25
Zweitwiegung: 27.05.2022 20:43

Brutto: H 19.580kg
Tara: H 10.340kg
Netto: E 9.340kg

Spediteur: Holzkontor Preußen GmbH **KFZ:** B-HK 2895

Position	Bezeichnung	AVV	Menge	Gewicht [kg]
1	Verpackungen aus Holz Bemerkung:	150103	t	4.220
2	Holz All-All Bemerkung:	170201	t	5.120

Unterschrift Anlieferer _____

Masswerte aus frei programmierbarer Zusatzanordnung. Die gesuchten Masswerte können eingesehen werden.

Holzkontor Preussen GmbH Sophienwerderweg 60 13597 Berlin Deutschland	Kontaktinformationen wagner@holzkontor.de 030 3330 2970 030 3330 2973	Bankverbindung Commerzbank AG BLAN: 0250 1004 0000 0207 7026 00 BIC: COBADE33HAN	Amtsgenoss Berlin HBR 85771 Ust-IdNr.: DE 235 098 982 Steuernr.: DE 249626460
---------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------

Abb. 5.24: Kundenportal Beispiel neuer Wiegebeleg

6 Projektabschluss und Fazit

Das nun folgende Fazit wird in zwei Teilbereiche getrennt. Beginnend kommt es zum Aufzeigen des aktuellen Projektstatus, darüber hinaus wird ein Ausblick auf eine darauf folgende Bachelorarbeit gegeben. Im Anschluss wird ein persönliches Fazit gezogen.

Nach zwölf Wochen, in denen sich umfangreich mit dem Thema DevOps, ERP-System und der Programmierung auseinandergesetzt wurde, obwohl kaum Erfahrung in diesen Bereichen bestanden, konnten schon viele Meilensteine auf dem Weg zum fertigen Kundenportal umgesetzt und neue Erfahrungen gesammelt werden, welche HKP definitiv neue Möglichkeiten eröffnen. Nichtsdestotrotz ist es noch ein weiter Weg zur vollständigen Implementierung des Kundenportals. Dieses Projekt dient dabei als Grundlage für die anschließende Bachelorarbeit, in der noch weitere Punkte umgesetzt werden sollen. Zum einen wird versucht, eine Optical Character Recognition (OCR) zu implementieren, um Lieferscheine von Kunden auslesen zu können und zum anderen gibt es noch weitere Optionen, wie z. B. den gesamten Waageprozess umzugestalten. Inwieweit dies jedoch umgesetzt werden kann, wird sich während der Bachelorarbeit herausstellen.

Mir persönlich hat dieses Projekt sehr viel Freude bereitet und ich konnte mich an vielen neuen umfangreichen und abwechslungsreichen Aufgaben probieren. Dabei fand ich besonders spannend, das erlernte Wissen aus den verschiedenen Modulen sinnvoll miteinander verknüpfen und nutzen zu können. Erst in diesem Semester habe ich so richtig begonnen in PHP zu programmieren und der Fortschritt, den ich dabei erzielen konnte, war enorm, was einen Schub für die Motivation zur Folge hatte. Der Aufbau des DevOps-Zyklus sowie die Nutzung der IDE PhpStorm und des Frameworks Laravel stellten dabei eine besondere Herausforderung dar, weil die Einarbeitung und das Einlesen besonders schleppend verliefen. Dieses schleppende Anlernen zeigte sich auch in diesem Dokument, denn es fiel mir grundsätzlich nicht leicht, einen derart langen zusammenhängenden Text zu formulieren und meine Aussagen anhand von Zitaten zu belegen. Besonders in den Abschnitten des DevOps und der IDE konnte ich dies persönlich feststellen.

Der Zugewinn an Erfahrung überwiegt dabei gegenüber den Widrigkeiten, die ich während der Projektphase hatte und ich denke, dass ich noch langfristig von dieser Projektphase profitieren werde.

Anhang

A Fragebögen

Wie erwähnt können die Fragebögen mit den jeweiligen Antworten der Fachabteilungen hier eingesehen werden.

Tab. A.1: Fragebogen – Annahme – Pamela Dziecol 11.04.2022

Frage	Antwort
1.	Auflistung der der Wiegebelege und der einzelnen Sorten/Positionen, Ausdruck als PDF, HKP soll bearbeiten können
2.	Kunden nur eigene Wiegebelege, HKP alle Wiegebelege
3.	Wiegebelegnummer, Sorte, Kunde, Baustelle, Auftragsnummer
4.	Kraftfahrer (eigene), Disponenten (alle), Buchhaltung und Betriebsleitung
5.	Wiegebeleg anlegen, bearbeiten und eine Tagesliste
6.	An dem aktuellen orientieren
7.	Siehe oben
8.	Ausdruck nur der eigenen Datensätze, dann ja
9.	Nur relevante Daten, also keine Schlüsselwerte, sondern eher Kundenname etc.

Tab. A.2: Fragebogen – Disposition – Reiner Uderhardt 12.04.2022

Frage	Antwort
1.	Anmelden/Beantragen von Container Tausch, Stellen, Abholen Anlegen von Lieferscheinen um als Selbstanlieferer Material anliefern zu können
2.	Aufstellung der aktuellen Container auf dem Gelände der Kunden, die Logik muss stimmen
3.	Baustellen und Sorten
4.	Disponent und Kraftfahrer, vielleicht Betriebsleiter je nach Größe des Betriebes und der Struktur (HKP Kraftfahrer mehr Berechtigung)
5.	Containeraktion, Containerplanung, vielleicht ein Kalender, um die Aufträge zuordnen zu können, im Kopf die Fahrzeuge und dann links die Uhrzeiten und die Fahraufträge reinschieben
6.	Nur für Lieferschein relevant, Darstellung wie Wiegebeleg nur angepasst an den Lieferschein
7.	Siehe oben
8.	Für HKP relevant, aber Kunden sollen das eher nicht sehen
9.	Je mehr desto besser, kann im Nachhinein ausgewertet werden

Tab. A.3: Fragebogen – Einkauf/Verkauf – Michael Ludwig 13.04.2022

Frage	Antwort
1.	Preise anzeige, suchen, anfragen und ausdrucken
2.	Nur eigene ganz wichtig, Preise dürfen niemals Kundenübergreifend ersichtlich sein!!!
3.	Sortennummer, Baustelle, da Preise für Baustellen unterschiedlich sein können
4.	Nur Vertriebler und Betriebsleiter
5.	Noch nicht sicher vielleicht für HKP anlegen
6.	Unser Kopfbogen mit den Daten zu den Sorten und Preisen
7.	Frage entfernt
8.	Nur eigene Daten dann ist der Export ok
9.	Weniger ist besser, nur der Preis, die Sorte und Baustelle und vielleicht die Laufzeit

Tab. A.4: Fragebogen – Buchhaltung – Karina Herzog 14.04.2022

Frage	Antwort
1.	Rechnungen müssen selbstständig rausgesucht werden können, als PDF ausdrucken, wenn möglich
2.	Rechnungen, wichtig, nicht jeder darf Zugriff auf Rechnungen haben
3.	Kunde, Adresse etc. Sorten Baustelle, Summe der Rechnung und Rechnungsdatum
4.	Buchhaltung oder Betriebsleitung, bei Dispositionsrechnung Dispo
5.	Eigentlich nur Rechnung suchen
6.	Siehe aktuelle Rechnung
7.	Frage entfernt
8.	Wenn die Kunden dann besser vergleichen können, wäre dies gut, weil dann weniger Anrufe kommen, ist auch wichtig fürs Steuerbüro
9.	Ähnlich wie bei der Suchfunktion

Tab. A.5: Fragebogen – Controlling – Heiko Schurwin 14.04.2022

Frage	Antwort
1.	Mehr Auswertungen sind immer gut, wir sind ein Dienstleister und bieten deshalb den besten Service, auch wichtig für Abfallstatistik etc. die Auflagen für Entsorgungsfachbetriebe sind sehr hoch
2.	Betriebsleitung alle, ansonsten nur für deren Bereich relevante Daten
3.	Eher welche Daten sollen angezeigt werden, wie soll der Aufbau sein
4.	Muss je nach Abteilung eingestellt werden, der eine soll mehr Auswertungen erhalten, der andere weniger
5.	Gesplittet nach Abteilung oder Funktion der Datensätze, welche Daten sind relevant für die Abfallstatistik, Auswertung über Sortierung etc. alle Daten, welche die Behörde einfordert in einen Bereich, ein klick alles runterladen
6.	Aufbereitet für die Behörde oder bei einer Prüfung oder Verlängerung der des Zertifikats als Entsorgungsfachbetrieb
7.	Frage entfernt
8.	Ja definitiv, HKP gibt diese auch in der Form bei der Behörde ab.
9.	Relevante Daten für die Behörde

B Mockups

Wie beschrieben sind die übrigen Mockups hier aufzufinden. Des Weiteren besteht die Möglichkeit, Zugriff auf das verbundene GitLab zu erhalten: <https://gitlab.bht-berlin.de/s79873/praxisprojekt>

Bausite	Adresse	PLZ	Ort	Containerszahl	offene Aufträge	erledigt
HKP	Sophienwerderweg 60	13577	Berlin	10	2	0
AIR	Sophienwerderweg 60	13577	Berlin	4	1	1

Abb. B.1: Mockup – Disposition

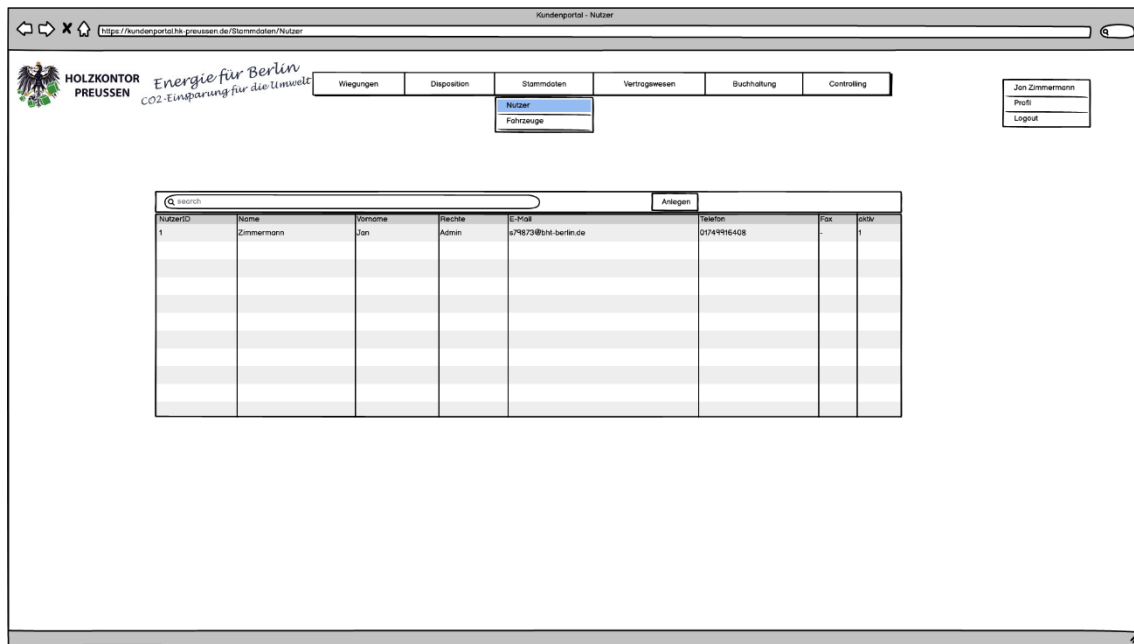


Abb. B.2: Mockup – Nutzer

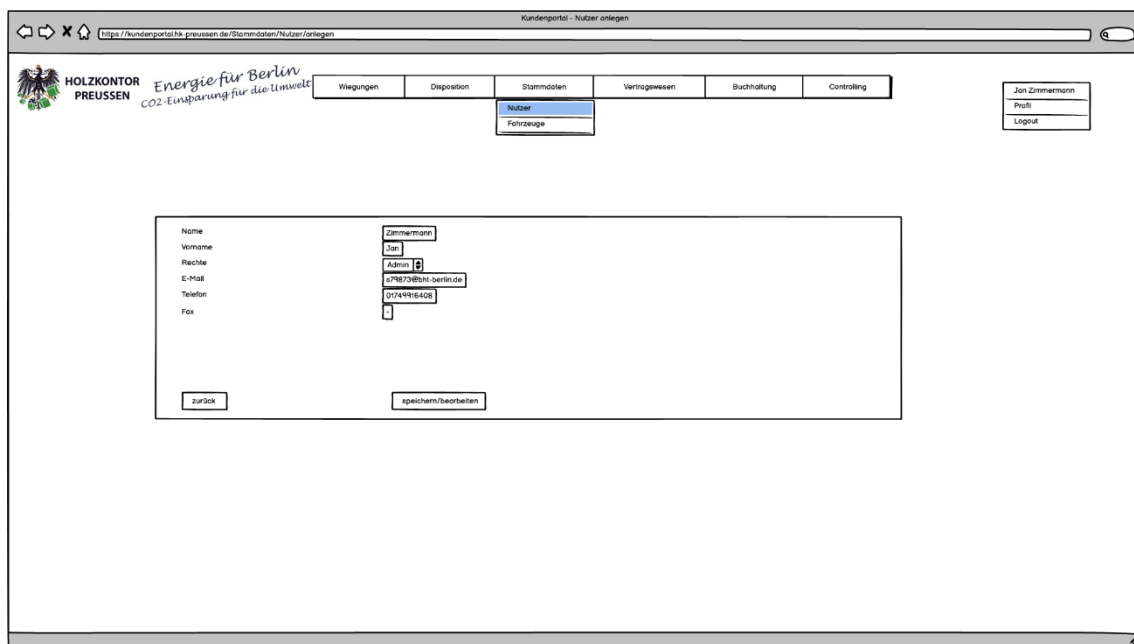


Abb. B.3: Mockup – Nutzer anlegen

Kundenportal - Fahrzeuge

https://kundenportal.hk-preussen.de/Stammdaten/Fahrzeuge

HOLZKONTOR PREUSSEN *Energie für Berlin*
CO₂-Einsparung für die Umwelt

Wiegungen Disposition Stammdaten Vertragsgewesen Buchhaltung Controlling

Nutzer
Fahrzeuge

Jan Zimmermann
Profil
Logout

Suche

FahrzeugID	Kennzeichen	Aktiv
1	B-JZ 2311	1

Abb. B.4: Mockup – Fahrzeuge

Kundenportal - Fahrzeuge

https://kundenportal.hk-preussen.de/Stammdaten/Fahrzeuge

HOLZKONTOR PREUSSEN *Energie für Berlin*
CO₂-Einsparung für die Umwelt

Wiegungen Disposition Stammdaten Vertragsgewesen Buchhaltung Controlling

Nutzer
Fahrzeuge

Jan Zimmermann
Profil
Logout

Kennzeichen B-JZ 2311

Zurück speichern/bearbeiten

Abb. B.5: Mockup – Fahrzeuge anlegen

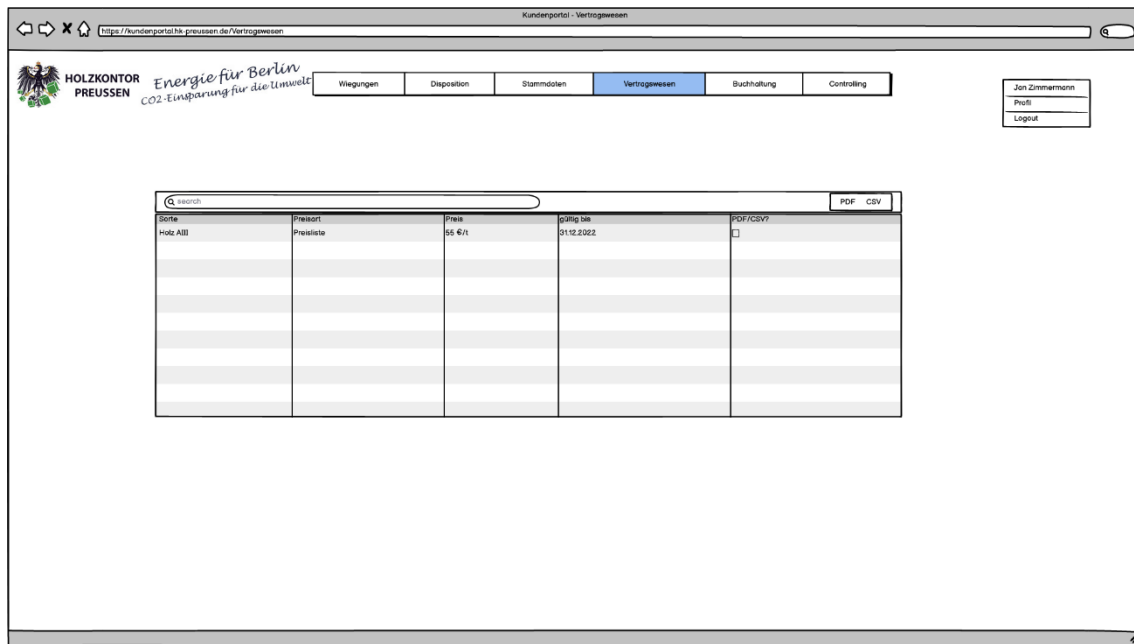


Abb. B.6: Mockup – Vertragswesen

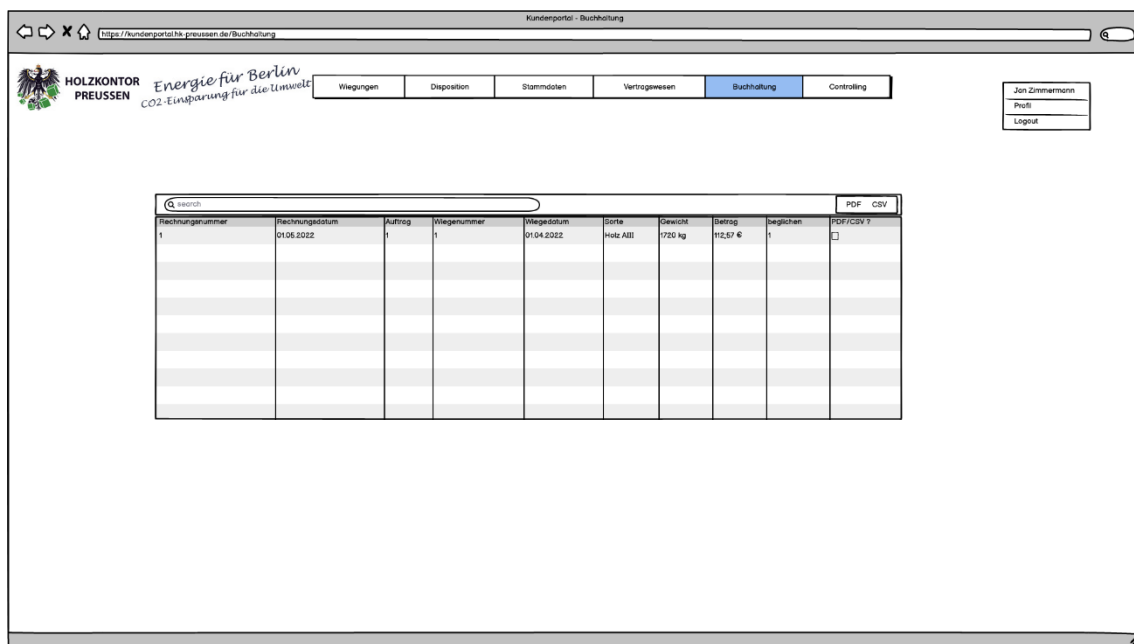



Abb. B.7: Mockup – Buchhaltung

C Kundenportal

Wie beschrieben findet nachfolgend eine nähere Auflistung des Kundenportals statt. Darüber hinaus werden einige Codebeispiele gegeben. Des Weiteren besteht die Möglichkeit, Zugriff auf das verbundene GitLab zu erhalten: <https://gitlab.bht-berlin.de/s79873/praxisprojekt>



Anmelden

Passwort

☐ Angemeldet bleiben

[Passwort vergessen?](#) **EINLOGGEN**

Abb. C.1: Kundenportal – Loginseite

Auf dem Dashboard wird angezeigt, was die einzelnen Module bewirken. Es soll im Rahmen der Bachelorarbeit zusätzlich die Möglichkeit geboten werden, dies individuell anzupassen, um vorab Informationen und Statistiken erhalten zu können.








 Holzkontor Preußen GmbH - Kundenportal		
 Waagemodul Abrufen von Wiegebelegen und dazugehöriger Dokumentation. Möglichkeit des Ein- und Auswiegens an einem Terminal.	 Disposition Erstellen von Anliefer- und Abholscheinen. Möglichkeit der Planung von Containergestellungen, Containerabholungen oder eines Containertausches. Des Weiteren Optionen der Tourenplanung und Fahrzeugüberwachung.	 Stammdatenpflege Anlegen von Kunden, Nutzern, Fahrzeugen, Sorten und alle weiteren relevanten Stammdaten.
 Vertragswesen Erstellen von Verträgen/Angeboten und Abrufen dieser. Möglichkeit der Preisgestaltung und -kontrolle.	 Buchhaltung Erstellung von Rechnungen und Mahnungen. Prüfung von Zahlungsein- und -ausgängen. Einbinden von externen Rechnungen.	 Controlling Abrufen von Statistiken und Auswertungen, um einen aktuellen Überblick über z.B. die Mengen oder Kosten und Margen zu erhalten.

Abb. C.2: Dashboard

Das Waagemodul dient der Erfassung der Warenein- und Warenausgänge. Es wird im rechten Bereich angezeigt, welche Fahrzeuge sich momentan auf dem Gelände befinden. Darüber hinaus können Wiegungeen angezeigt und gedruckt werden.

Waage → Erstwiegung

ERSTWIEGUNG

Lieferschein

Kennzeichen

Spediteur

Kunde

Baustelle

Sorte

Gewicht Erstwiegung

Datum Erstwiegung

Abbrechen

Waage Speichern

HOFLISTE

KENNZEICHEN

KUNDE

BAUSTELLE

SORTE

ERSTELLT AM

Suche ...

TT.

Abb. C.3: Waage – Wiegung anlegen

Im Dispositionsmodul können Container gepflegt und Aufträge erstellt werden. Über diese Aufträge kann eine Lieferung und Wiegung erfolgen.

Disposition → Container

Hinzufügen

CSV

CONTAINERNUMMER	VOLUMEN	UVV	GEWICHT	DECKEL	ERSTELLT VON	ERSTELLT AM
1	40	2024-06-13	2640	0	Jan Zimmermann	22.06.2022

Abb. C.4: Disposition – Container pflegen

Disposition → Lieferschein

Hinzufügen

CSV

PDF

LIEFERSCHEINNUMMER	KUNDE	BAUSTELLE	SORT	ERSTELLT VON	ERSTELLT AM
↓ 1	Remondis Berlin GmbH	Keine Baustelle	Verpackungen aus Holz	Jan Zimmermann	22.06.2022
↓ 2	Holzkontor Preußen GmbH	Keine Baustelle	Holz All-All	Jan Zimmermann	22.06.2022

Abb. C.5: Disposition – Lieferschein abrufen

Die Ansicht der Nutzer ist hierbei beispielhaft für die Stammdaten angegeben. Diese Ansichten finden sich auch bei den übrigen Unterseiten.

NUTZERID	VORNAME	NACHNAME	ERSTELLT VON	ERSTELLT AM
1	Jan	Zimmermann	Jan Zimmermann	22.06.2022

Abb. C.6: Stammdaten – Nutzer anzeigen

Abb. C.7: Stammdaten – Nutzer erstellen

Abb. C.8: Stammdaten – Nutzer bearbeiten

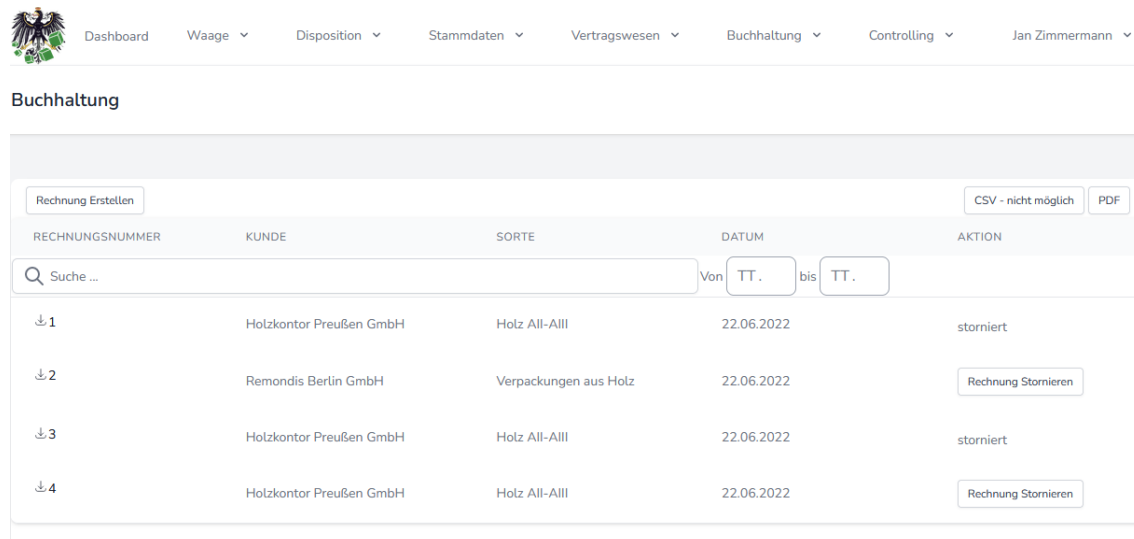
Eine Ausnahme hierbei bilden die Kunden, da zu jedem Kunden Baustellen, Nutzer und Kennzeichen zugeordnet sind. Wenn also ein Kunde bearbeitet wird, so findet auch eine Auflistung der jeweiligen Baustellen, Nutzern und Kennzeichen statt, welche hierbei wiederum bearbeitet und angelegt werden können.

Abb. C.9: Stammdaten – Kunden bearbeiten

Im Bereich Vertragswesen können Preise gepflegt werden. Dies geschieht zumeist über Preislisten, da Kundengruppen identische Preise erhalten und diese schnell auf Marktveränderungen angeglichen werden können. Des Weiteren besteht die Möglichkeit Individualpreise für Kunden und Baustellen zu hinterlegen.

Abb. C.10: Vertragswesen – Preisliste abrufen

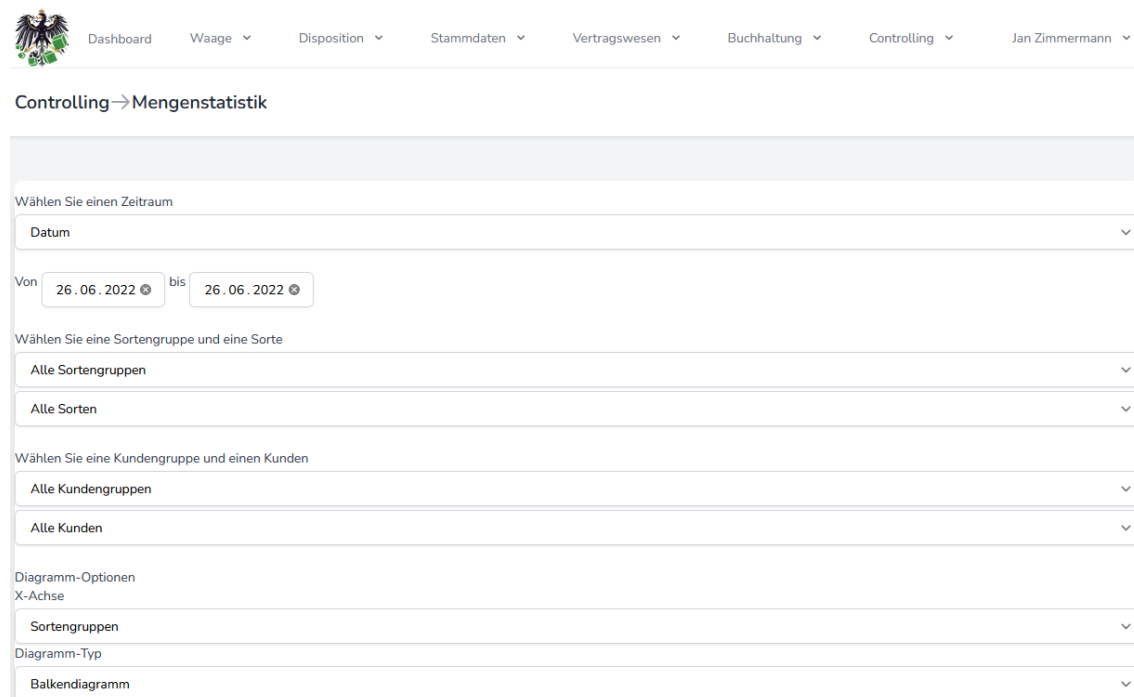
Im Buchhaltungsmodul können Rechnungen per Knopfdruck automatisch erstellt werden. Zusätzlich ist es möglich, sich die Rechnungen anzeigen zu lassen oder diese zu stornieren, um Änderungen vornehmen zu können.



RECHNUNGSNUMMER	KUNDE	SORTE	DATUM	AKTION
1	Holzkontor Preußen GmbH	Holz AII-AIII	22.06.2022	storniert
2	Remondis Berlin GmbH	Verpackungen aus Holz	22.06.2022	Rechnung Stornieren
3	Holzkontor Preußen GmbH	Holz AII-AIII	22.06.2022	storniert
4	Holzkontor Preußen GmbH	Holz AII-AIII	22.06.2022	Rechnung Stornieren

Abb. C.11: Buchhaltung – Rechnungen abrufen/erstellen/stornieren

Das Controllingmodul ist vorbereitet, sodass Echtzeitauswertungen über die Materialbewegungen erledigt werden können. Dabei besteht die Möglichkeit, eine Vorauswahl zu treffen, woraufhin sich der Graph ändert.



Controlling → Mengenstatistik

Wählen Sie einen Zeitraum

Datum

Von 26.06.2022 bis 26.06.2022

Wählen Sie eine Sortengruppe und eine Sorte

Alle Sortengruppen

Alle Sorten

Wählen Sie eine Kundengruppe und einen Kunden

Alle Kundengruppen

Alle Kunden

Diagramm-Optionen

X-Achse

Sortengruppen

Diagramm-Typ

Balkendiagramm

Abb. C.12: Controlling – Auswahl zur Statistik

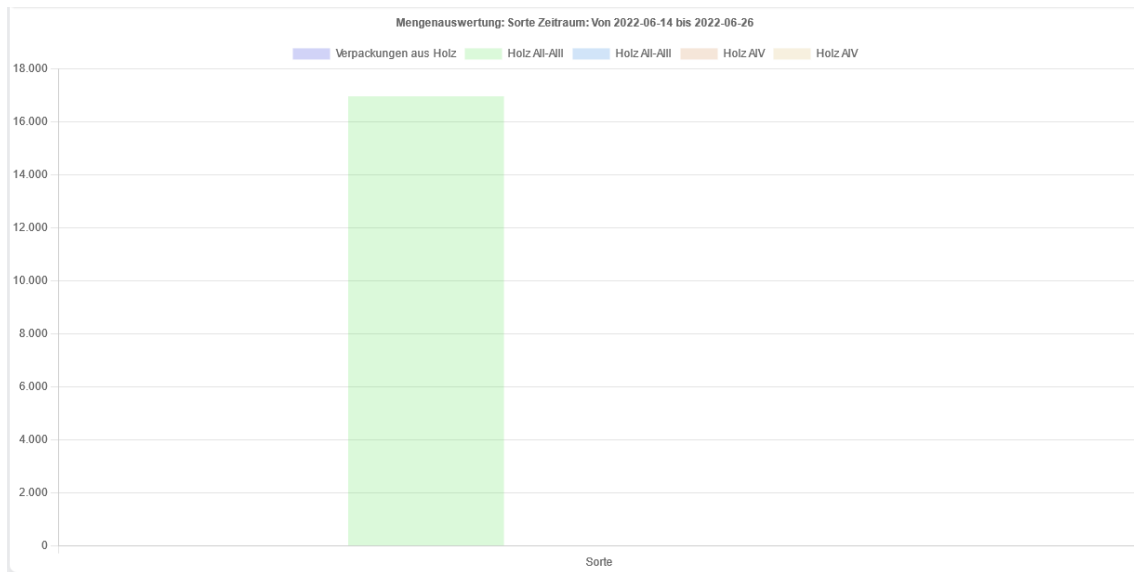


Abb. C.13: Controlling – Graph exemplarisch nach Sorten

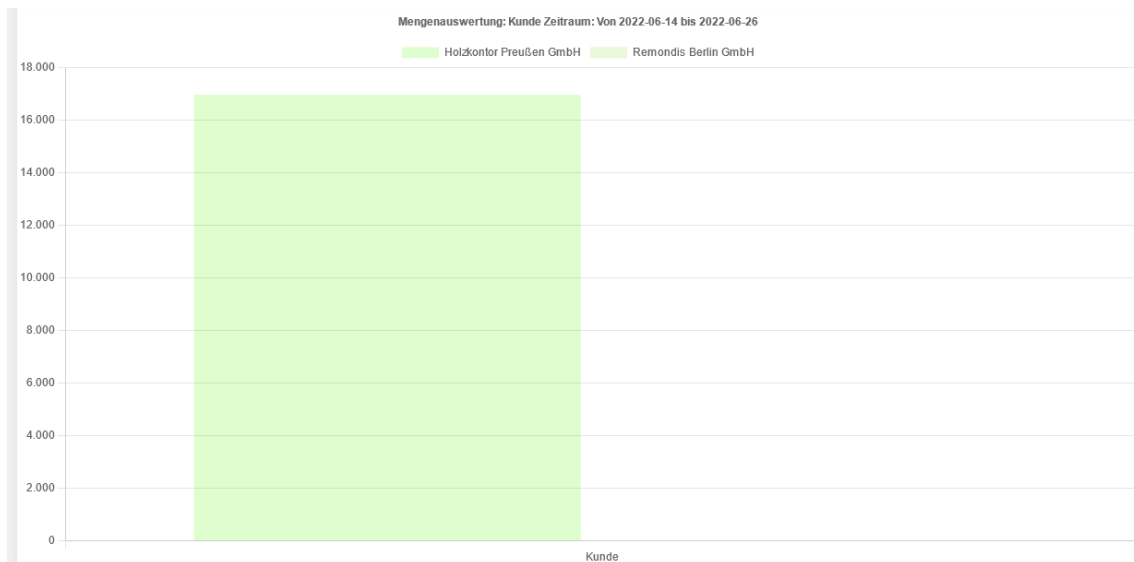


Abb. C.14: Controlling – Graph exemplarisch nach Kunden

Im Anhang soll nun noch auf die Programmierung der einzelnen Komponenten eingegangen werden.

Jeder Tabelleneintrag in der Datenbank hat ein Pendant in den Models. Zum Zeitpunkt der Dokumentenabgabe befinden sich in der Datenbank 58 Tabellen. Dabei wurden speziell Gegebenheiten der Gesetzgebung berücksichtigt. Alleine die Tabellen für die Abfallsorten, welche durch das Abfallrecht vorgegeben werden, beinhalten 843 Einträge für Abfalltypen, 111 Einträge für Abfallgruppen und 20 Einträge für Abfallklassen (*Umweltbundesamt - european waste catalogue*, 2016). Darüber hinaus wurden auch Standards für die Materialverbringung implementiert, wie die Umsetzung der Incoterms. Dies sind Handelsklauseln mit internationaler Gültigkeit, die von vielen Unternehmen genutzt werden (*Hellmann East Europe - Incoterms® 2020*, 2022).

SELECT * FROM `varieties`

☐ Messen ☐ [Inline bearbeiten] ☐ [Bearbeiten] ☐ [SQL erklären] ☐ [PHP-Code erzeugen] ☐ [Aktualisieren]

☐ Alles anzeigen | Anzahl der Datensätze: 25 | Zeilen filtern: Diese Tabelle durchsucht | Nach Schlüssel sortieren: keine

+ Optionen

			id	varietygroup_id	AVV	name	created_at	updated_at	created_by	updated_by	active
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	1	1	150103	Verpackungen aus Holz	2022-05-27 19:02:11	2022-05-27 19:02:11	1	1
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	2	1	170201	Holz AII-AIII	2022-05-27 19:02:11	2022-05-27 19:02:11	1	1
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	3	1	200138	Holz AII-AIII	2022-05-27 19:02:11	2022-05-27 19:02:11	1	1
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	4	1	170204	Holz AIV	2022-05-27 19:02:11	2022-05-27 19:02:11	1	1
<input type="checkbox"/>	Bearbeiten	Kopieren	Löschen	5	1	200137	Holz AIV	2022-05-27 19:02:11	2022-05-27 19:02:11	1	1

Abb. C.15: Tabelle – Sorten

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Variety extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'AVV',
        'varietygroup_id',
        'created_by',
        'updated_by',
        'active'
    ];

    public function getcreator()
    {
        return $this->belongsTo(User::class, 'created_by');
    }

    public function getgroup()
    {
        return $this->belongsTo(Varietygroup::class,
```

```
'varietygroup_id');
    }
}
```

Es wird hierbei explizit aufgezeigt, welche Beziehungen zu anderen Tabellen/Models bestehen.

Für das Bearbeiten und Anlegen einer Sorte wird genau eine Komponente und eine Blade-Datei benötigt. Dies ist typisch für ein Framework, dass nach dem Model-View-Controller Konzept aufgebaut ist.

```
<?php

namespace App\Http\Livewire>Maindata\Edit;

use App\Models\Variety;
use App\Models\Varietygroup;
use Illuminate\Support\Facades\Auth;
use Livewire\Component;

class VarietyEditComponent extends Component
{
    public $values;
    public $valuesID;
    public $submitAction;
    public $name = '';
    public $AVV = '';
    public $varietygroup_id = 1;
    public $varietygroup;
    public $active = 1;

    protected $rules = [
        'name' => 'filled',
        'AVV' => 'filled',
    ];

    protected $listeners = ['updateVarietygroup'];

    public function updateVarietygroup()
    {
        return redirect(request()->header('Referer'));
    }

    public function mount(){
        $this->varietygroup = Varietygroup::all();
        if ($this->values) {
            $this->valuesID = $this->values->id;
            $this->name = $this->values->name;
            $this->AVV = $this->values->AVV;
            $this->varietygroup_id = $this->values->varietygroup_id;
        }
    }

    public function submit()
    {
        if ($this->submitAction == "create") {
            $this->validate();
            $values = Variety::create([
                'name' => $this->name,
```

```

        'AVV' => $this->AVV,
        'varietygroup_id' => $this->varietygroup_id,
        'created_by' => Auth::user()->id,
        'updated_by' => Auth::user()->id,
        'active' => $this->active,
    ];
    $values->save();
    redirect(route('variety'));
} elseif ($this->submitAction == "update") {
    $this->validate();
    $values = Variety::find($this->valuesID);
    $values->name = $this->name;
    $values->AVV = $this->AVV;
    $values->varietygroup_id = $this->varietygroup_id;
    $values->updated_by = Auth::user()->id;
    $values->active = $this->active;
    $values->save();
    redirect(route('variety'));
}

}

public function render()
{
    return view('livewire.maindata.edit.variety-edit-component');
}
}

```

```

<div>
    @livewire('maindata.edit.create.varietygroup-component')
    <div class="space-y-8 divide-y divide-gray-200">
        <div class="space-y-8 divide-y divide-gray-200">
            <div>
                <div class="mt-6 grid grid-cols-1 gap-y-6 gap-x-4 sm:grid-cols-6">
                    <div class="sm:col-span-6">
                        <label for="name" class="block text-sm font-medium text-gray-700"> {{ __('Variety') }} </label>
                        <div class="mt-1 flex rounded-md shadow-sm">
                            <input wire:model.defer="name" type="text"
                                name="name" id="name"
                                autocomplete="name"
                                class="shadow-sm focus:ring-indigo-500 focus:border-indigo-500 block w-full sm:text-sm border border-gray-300 rounded-md">
                        </div>
                        <x-jet-input-error for="name" class="mt-2"/>
                    </div>
                    <div class="sm:col-span-6">
                        <label for="AVV" class="block text-sm font-medium text-gray-700"> AVV </label>
                        <div class="mt-1 flex rounded-md shadow-sm">
                            <input wire:model.defer="AVV" type="text"
                                name="AVV" id="AVV"
                                autocomplete="AVV"
                                class="shadow-sm focus:ring-indigo-500 focus:border-indigo-500 block w-full sm:text-sm border border-gray-300 rounded-md">
                        </div>
                        <x-jet-input-error for="AVV" class="mt-2"/>
                    </div>
                </div>
            </div>
        </div>
    </div>

```



```

        <div class="sm:col-span-6">
            <div class="flex">
                <button wire:click="$emit('openModalVarietygroup')" class="border-solid border-2 border-gray-300 rounded">
                    <svg
xmlns="http://www.w3.org/2000/svg" class="h-4 w-4" fill="none"
viewBox="0 0 24 24"
                        stroke="currentColor">
                            <path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M12 4v16m8-8H4"/>
                        </svg>
                    </button>
                    <label for="varietygroup_id" class="block text-sm font-medium text-gray-700"> {{ __('Varietygroup') }} </label>
                </div>
                <div class="mt-1 flex rounded-md shadow-sm">
                    <select wire:model.defer="varietygroup_id"
id="varietygroup_id" name="varietygroup_id"
                        class="shadow-sm focus:ring-indigo-500 focus:border-indigo-500 block w-full sm:text-sm border border-gray-300 rounded-md">
                        @foreach ($varietygroup as $group)
                            <option id="{{ $group->id }}" value="{{ !! $group->id !! }}">{{ $group->name }}</option>
                        @endforeach
                    </select>
                </div>
                <x-jet-input-error for="varietygroup_id"
class="mt-2"/>
            </div>
        </div>
    </div>
</div>
<div class="pt-5">
    <div class="flex justify-start">
        <a type="button" href="{{ route('variety') }}"
            class="bg-white py-2 px-4 border border-gray-300 rounded-md shadow-sm text-sm font-medium text-gray-700 hover:bg-gray-50 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500">
                {{ __('Cancel') }}
            </a>
            <button disabled wire:loading type="button"
                class="ml-3 inline-flex justify-center py-2 px-4 border border-transparent shadow-sm text-sm font-medium rounded-md text-white bg-indigo-600 hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500">
                    {{ __('Please wait ...') }}
                </button>
                <button wire:click="submit" wire:loading.remove
type="button"
                    class="ml-3 inline-flex justify-center py-2 px-4 border border-transparent shadow-sm text-sm font-medium rounded-md text-white bg-indigo-600 hover:bg-indigo-700 focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500">
                        {{ __('Variety') }} {{ __('Save') }}
                    </button>
                </div>
    </div>
</div>
</div>

```

Diese Beispiele für den Code können im GitLab näher beleuchtet werden. Die einzelnen Komponenten sind immer ähnlich aufgebaut. Exemplarisch habe ich eine Komponente und eine Blade-Datei aufgezeigt. Es gibt aber noch wesentlich mehr Dateien, die sich rein mit den Sorten befassen. Je Unterseite gibt es ca. sieben Dateien, um alle Ansichten, von der Übersicht in der Tabelle, über das Anlegen bis hin zur Bearbeitung darzustellen.

Literaturverzeichnis

- Adamu, J., Hamzah, R., & Rosli, M. M. (2020). Security issues and framework of electronic medical record: A review. *Bulletin of Electrical Engineering and Informatics*, 9(2). <https://doi.org/10.11591/eei.v9i2.2064>
- Alt, R., Auth, G., & Kögler, C. (2017). *Innovationsorientiertes IT-Management mit DevOps*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-18704-0>
- Cepuc, A., Botez, R., Craciun, O., Ivanciu, I.-A., & Dobrota, V. (2020). Implementation of a Continuous Integration and Deployment Pipeline for Containerized Applications in Amazon Web Services Using Jenkins, Ansible and Kubernetes. *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 1–6. <https://doi.org/10.1109/RoEduNet51892.2020.9324857>
- Dobslaw, F., Persson, T., & Wikhög, M. (2018). *MID SWEDEN UNIVERSITY*. 48.
- Eltahawey, A. O. (2016). *Hyper Text Markup Language HTML: A Tutorial*. 26.
- Halstenberg, J., Pfitzinger, B., & Jestädt, T. (2020). *DevOps: Ein Überblick*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-31405-7>
- Hellmann East Europe—Incoterms® 2020*. (2022, Juni 4). [Betrieb]. Hellmann East Europe - Incoterms. <https://www.hellmann-east-europe.com/hellmannpedia/i/incoterms/>
- Holzkontor Preussen GmbH - AI*. (2022, Juni 4). [Betrieb]. Holzkontor Preussen GmbH. <https://www.hk-preussen.de/wp-content/uploads/2019/05/alholzklasse-a1.jpg>
- Holzkontor Preussen GmbH - AIV*. (2022, Juni 4). [Betrieb]. Holzkontor Preussen GmbH. <https://www.hk-preussen.de/wp-content/uploads/2019/05/alholzklasse-a4.jpg>

- Holzkontor Preussen GmbH - Anlage.* (2022, Juni 4). [Betrieb]. Holzkontor Preussen GmbH. <https://www.hk-preussen.de/wp-content/uploads/2019/05/foerderbaender-halle.jpg>
- Holzkontor Preussen GmbH - Sennebogen.* (2022, Juni 4). [Betrieb]. Holzkontor Preussen GmbH. <https://www.hk-preussen.de/wp-content/uploads/2021/03/Sennebogen.jpg>
- Katal, A., Bajoria, V., & Dahiya, S. (2019). DevOps: Bridging the gap between Development and Operations. *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 1–7. <https://doi.org/10.1109/ICCMC.2019.8819631>
- Laaziri, M., Benmoussa, K., Khouliji, S., & Kerkeb, M. L. (2019). A Comparative study of PHP frameworks performance. *Procedia Manufacturing*, 32, 864–871. <https://doi.org/10.1016/j.promfg.2019.02.295>
- Nordhausen, T., & Hirt, J. (2020, Oktober 13). *Manual zur Literaturrecherche in Fachdatenbanken* [Publikation]. Manual zur Literaturrecherche in Fachdatenbanken. https://www.researchgate.net/profile/Julian-Hirt/publication/345700600_RefHunter_Manual_zur_Literaturrecherche_in_Fachdatenbanken_Version_50/links/5fab1f49a6fdcc331b94483a/RefHunter-Manual-zur-Literaturrecherche-in-Fachdatenbanken-Version-50.pdf
- Pekša, J. (2018). Extensible Portfolio of Forecasting Methods for ERP Systems: Integration Approach. *Information Technology and Management Science*, 21, 64–68. <https://doi.org/10.7250/itms-2018-0010>
- Rad, B. B., Bhatti, H. J., & Ahmadi, M. (2017). *An Introduction to Docker and Analysis of its Performance*. 9.

- Schoeneberg, K.-P. (2011). *Kritische Erfolgsfaktoren von IT-Projekten. Eine empirische Analyse von ERP-Implementierungen am Beispiel der Mineralölbranche*. Rainer Hampp Verlag. <https://doi.org/10.1688/9783866187122>
- Than, P. P., & Phyu, M. P. (2019). Continuous integration for Laravel applications with GitLab. *Proceedings of the International Conference on Advanced Information Science and System*, 1–6. <https://doi.org/10.1145/3373477.3373479>
- Töpfer, A. (2020). *Strategische Positionierung und Kundenzufriedenheit: Anforderungen – Umsetzung – Praxisbeispiele*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-32019-5>
- Umweltbundesamt—European waste catalogue. (2016, April 1). Umweltbundesamt - european waste catalogue. <https://www.umweltbundesamt.de/sites/default/files/medien/2503/dokumente/2014-955-eg-en.pdf>
- Vadivelu, K., Balaji, N., Poongavanam, N., Tamilselvan, S., & Rajakumar, R. (2018). *CLOUD - ERP: IMPLEMENTATION STRATEGIES, BENEFITS AND CHALLENGES*. 6.
- Valdivia, R. G. B. (2019). *Collaborative Learning Using Git with GitLab in Students of the Engineering Programming Course*. 10.

Abschließende Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbstständig, ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Berlin, den 30. Juni 2022