



Abschlussprüfung Sommer 2021

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

JuFi Portal

SharePoint Lösung für die Geberit Junior Firma,

basierend auf dem SharePoint Framework

Abgabetermin: 01.06.2021

Prüfungsbewerber:

Jona Christian Sorg

Gisela-Franke-Weg 4

88630 Pfullendorf



Ausbildungsbetrieb:

Geberit Verwaltungs GmbH

Theuerbachstraße 1

88630 Pfullendorf

Inhaltsverzeichnis

Abbildungsverzeichnis.....	II
Tabellenverzeichnis.....	III
Abkürzungsverzeichnis.....	III
1 Einleitung	1
1.1 Projektbeschreibung.....	1
1.2 Projektziel.....	1
1.3 Projektumfeld	2
1.4 Projektbegründung.....	2
2 Projektplanungsphase	2
2.1 Ist-Analyse	2
2.2 Soll-Konzept.....	3
2.3 Projektphasen	3
2.4 Ressourcenplanung	3
2.5 Projektkosten	4
2.6 Amortisationsdauer	5
2.7 Qualitätsanforderungen.....	6
3 Durchführungsphase	6
3.1 Zielplattform	6
3.2 Klassendiagramme.....	6
3.3 SharePoint Seite und Microsoft Teams erstellen und einrichten	7
3.4 Online Shop Webpart Programmierung.....	7
3.5 Power Automate Flows.....	8
3.6 Hardwareverkauf Webpart Programmierung	9
3.7 Testing / Code-Review	9
3.8 Einrichtung	9
4 Projektabschlussphase.....	10
4.1 Abnahme.....	10
4.2 Dokumentation	10
4.3 Soll-Ist-Vergleich	11
4.4 Ausblick.....	11
4.5 Lessons Learned.....	12
4.6 Fazit	12
Quellenverzeichnis.....	13

A Anhang.....	i
A.1 Ausführliche Zeitplanung.....	i
A.2 Verwendete Ressourcen.....	ii
A.3 Amortisation.....	ii
A.4 JuFi Portal Product Backlog.....	iii
A.5 Klassendiagramm JuFi Portal.....	iii
A.6 Klassendiagramm Hardwareverkauf.....	iv
A.7 Geberit Junior Firma Team.....	iv
A.8 Artikelübersicht.....	v
A.9 Dialog – Artikel nicht verfügbar.....	v
A.10 Dialog – ungültige Menge.....	vi
A.11 Screenshot Warenkorb.....	vi
A.12 Bestellabschluss.....	vii
A.13 Screenshots Code	vii
A.14 Power Automate Flow.....	x
A.15 Hardwareverkauf Maske.....	xi
A.16 Berechtigungskonzept Listen.....	xi
A.17 Auszug Benutzerhandbuch.....	xi

Abbildungsverzeichnis

Abbildung 1: Amortisationsdiagramm.....	ii
Abbildung 2: Azure DevOps Product Backlog.....	iii
Abbildung 3: JuFi Portal Klassendiagramm.....	iii
Abbildung 4: Hardwareverkauf Klassendiagramm.....	iv
Abbildung 5: JuFi Teams.....	iv
Abbildung 6: Online-Shop Artikelübersicht.....	v
Abbildung 7: Artikel nicht verfügbar Dialog.....	v
Abbildung 8: Menge ungültig Dialog.....	vi
Abbildung 9: Warenkorb.....	vi
Abbildung 10: Bestellabschluss.....	vii
Abbildung 11: Auszug ArticleOverview.tsx.....	vii
Abbildung 12: Auszug WarenkorbOverview.tsx.....	viii
Abbildung 13: Auszug BestellAbschluss.tsx.....	ix
Abbildung 14: Power Automate Flow.....	x
Abbildung 15: Hardwareverkauf Anmeldemaske.....	xi

Tabellenverzeichnis

Tabelle 1: Grobe Zeitplanung.....	3
Tabelle 2: Kostenaufstellung.....	4
Tabelle 3: Zeiteinsparung.....	5
Tabelle 4: Soll-Ist-Vergleich.....	10
Tabelle 5: Zeitplanung.....	i

Abkürzungsverzeichnis

JuFi	Geberit Junior Firma
GIN	Geberit Intranet
SPFx	SharePoint Framework
MS	Microsoft
O365	Microsoft Office365
VS	Visual Studio
DEV	Development
SP	SharePoint

1 Einleitung

Die folgende Projektdokumentation schildert den Ablauf des IHK-Abschlussprojektes, welches ich im Rahmen meiner Ausbildung zum Fachinformatiker Fachrichtung Anwendungsentwicklung durchgeführt habe. Ausbildungsbetrieb ist die Geberit Verwaltungs GmbH am Standort Pfullendorf. Zurzeit beschäftigt die Geberit 1.570 Mitarbeiter am Standort Pfullendorf. Zu den Produkten des Unternehmens zählen neben Installations- und Rohrleitungssysteme auch bspw. Keramikprodukte.

1.1 Projektbeschreibung

Für die Auszubildenden im zweiten Lehrjahr hat die Ausbildungsleitung der Geberit eine Übungsfirma namens „Geberit Junior Firma“ (im folgenden JuFi genannt) gegründet. Die JuFi verkauft Produkte wie Schüsseln, Tassen und Vesperboxen und veranstaltet Sonderaktionen wie Hardwareverkäufe und Waffelverkäufe an Ostern und Weihnachten. Die Webseite sowie der Online Shop der JuFi befinden sich aktuell auf einer externen Seite eines Drittanbieters. Für die Hardwareverkäufe wird ein Formular auf einer SharePoint Seite genutzt. Der Datenaustausch findet über ein lokales Laufwerk statt und die Mitarbeiter kommunizieren außerhalb der JuFi Sitzungen nur über E-Mail.

1.2 Projektziel

Das Ziel des Projektes ist die JuFi-Webseite, den Online Shop, sowie das Hardwareverkauf-Formular abzulösen und intern auf einer SharePoint Lösung zu realisieren, welche dann direkt vom Geberit Intranet (im folgenden GIN genannt) aufrufbar sein wird. Für den Online Shop wird ein SharePoint Webpart (im folgenden SPFx¹ Webpart genannt) programmiert und mit Microsoft Power Automate Flows wird eine automatische Bestellbestätigung für die Mitarbeiter und eine Bestellbenachrichtigung für die JuFi generiert. In einer SharePoint-Liste kann die JuFi die bearbeiteten und neuen Bestellungen abrufen. Außerdem wird der JuFi ein Team in MS Teams zur Verfügung gestellt, welches sie als Kommunikationsweg nutzen können. Dort finden sie auch die Ordnerstruktur von dem lokalen Laufwerk wieder.

¹ „Das SharePoint Framework (SPFx) ist ein Seiten- und Webpart-Modell, das vollständige Unterstützung für die clientseitige SharePoint-Entwicklung, einfache Integration in SharePoint-Daten und Erweiterung von Microsoft Teams bietet.“
(siehe <https://docs.microsoft.com/de-de/sharepoint/dev/spfx/sharepoint-framework-overview>)

1.3 Projektumfeld

Auftraggeber des Projektes ist die Ausbildungsleitung und die Mitarbeiter der JuFi. Das Projekt wird im IT-Hub der Geberit Verwaltungs GmbH in Pfullendorf durchgeführt. Die benötigte Hard- und Software, sowie Lizenzen, stehen im Betrieb zur freien Verfügung.

1.4 Projektbegründung

Ein Nachteil an der aktuellen Online-Shop Lösung des Drittanbieters ist, dass die JuFi nur maximal 10 offene Bestellungen haben darf. Außerdem ist der Online Shop öffentlich und nur durch ein Passwort geschützt, welches im alten GIN für jeden Mitarbeiter frei zugänglich ist. Mit der SharePoint Lösung kann die JuFi mehr als 10 offene Bestellungen gleichzeitig betreuen und der Online Shop ist ohne Passwort für jeden Geberit Mitarbeiter frei zugänglich. Die Authentifizierung findet dabei über Single Sign-On statt. Ein weiteres Problem ist, dass die JuFi Mitarbeiter keine Übersicht über die Bestellungen haben. Die SharePoint Lösung bietet eine Liste mit Bestellungen sowie eine Liste mit bestellten Artikeln. Außerdem findet die Kommunikation der Mitarbeiter außerhalb der Sitzungen selten und wenn, dann nur über E-Mail statt. Mit einem Team in MS Teams können die Mitarbeiter chatten und wichtige Events auch außerhalb der Sitzungen planen und besprechen.

2 Projektplanungsphase

2.1 Ist-Analyse

Wie schon in 1.1 (Projektbeschreibung) erwähnt, befinden sich die JuFi Webseite und der Online-Shop aktuell noch auf einer externen Seite eines Drittanbieters. Die offenen Bestellungen sind auf 10 Stück begrenzt und auf den Online-Shop kann nur mit einem Passwort, welches im alten GIN steht, zugegriffen werden. Eine Übersicht mit offenen und abgeschlossenen Bestellungen gibt es nicht. Die Anmeldung für die Hardwareverkäufe finden über ein Formular statt, welches von der JuFi als GIN-Eintrag gepostet werden muss. Die Anmeldungen werden in einer SharePoint Liste gespeichert.

2.2 Soll-Konzept

Eine interne SharePoint Lösung soll die Webseite, den Online-Shop und das Hardwareformular in Form eines Portals ablösen. Außerdem sollen auf der SharePoint Lösung die Listen „Bestellungen“, „Bestellte Artikel“ und „Hardwareverkauf Anmeldungen“ abrufbar sein. Die JuFi soll ein Team in MS Teams zur Verfügung gestellt bekommen, auf dem sie Ihre Ordnerstruktur und die Listen wiederfinden. Das Team dient außerdem zur Verbesserung der Kommunikation innerhalb der JuFi.

2.3 Projektphasen

Für das Projekt waren 70 Stunden vorgesehen. Vor Projektbeginn wurden diese auf verschiedene Phasen verteilt. Daraus ist eine grobe Zeitplanung entstanden, welche in Tabelle 1 erfasst ist.

Projektphase	Geplante Zeit
Analysephase	3 h
Durchführungsphase	35,5 h
Implementierungsphase	12,5 h
Übergabe	4 h
Erstellen der Dokumentation	15 h
Gesamt	70 h

Tabelle 1: Grobe Zeitplanung

Die einzelnen Hauptphasen können noch in kleinere Unterpunkte zerlegt werden. Eine ausführliche Übersicht dieser Phasen befindet sich im Anhang A.1: Ausführliche Zeitplanung auf Seite i.

2.4 Ressourcenplanung

Eine Übersicht mit allen eingesetzten Ressourcen befindet sich im Anhang A.2: Verwendete Ressourcen auf Seite ii. Die Übersicht beinhaltet neben Hardware- und Softwareressourcen auch das Personal.

2.5 Projektkosten

Die Projektkosten die im Folgenden kalkuliert werden, berücksichtigen sowohl Personalkosten für die Entwicklung, als auch Ressourcenkosten (Hard- und Software, Büroarbeitsplatz, etc.). Um das Herausgeben genauerer Personalkosten zu vermeiden, wird mit einem Monatslohn von 1000€ Brutto für einen Auszubildenden gerechnet. Mit 13 Monatsgehältern, 30 Urlaubstagen und 253 Arbeitstagen im Jahr 2021, wird der Stundenlohn wie folgt berechnet:

$$7.5 \text{ h/Tag} \cdot 223 \text{ Tage/Jahr} = 1672.5 \text{ h/Jahr} \quad (1)$$

$$1000 \text{ €/Monat} \cdot 13 \text{ Monate/Jahr} = 13000 \text{ €/Jahr} \quad (2)$$

$$\frac{13000 \text{ €/Jahr}}{1672.5 \text{ h/Jahr}} \approx 7,77 \text{ €/h} \quad (3)$$

Der Stundenlohn eines Auszubildenden beträgt demnach 7,77€. Die Durchführung des Projektes beträgt 70 Stunden. Für die Ressourcennutzung wird ein pauschaler Stundensatz von 15€ angenommen. Um auch bei den anderen Mitarbeitern das Herausgeben genauerer Personalkosten zu vermeiden, wird mit einem pauschalen Stundenlohn von 25€ pro Mitarbeiter gerechnet. Die Kosten der einzelnen Vorgänge des Projektes und die Gesamtkosten lassen sich der Tabelle 2: Kostenaufstellung entnehmen.

Vorgang	Mitarbeiter	Zeit	Personal ²	Ressourcen ³	Gesamt
Entwicklungskosten	1x Auszubildender	70 h	543,90€	1.050,00€	1.593,90€
Abstimmungen	2x Auszubildende	2 h	31,08€	60,00€	91,08€
Code Review	2x Mitarbeiter	2 h	100,00€	60,00€	160,00€
Projektkosten gesamt					1.844,98€

Tabelle 2: Kostenaufstellung

² Personalkosten = Anzahl Mitarbeiter • Zeit • Stundensatz

³ Ressourcenkosten = Anzahl Mitarbeiter • Zeit • 15€

2.6 Amortisationsdauer

Um herauszufinden ab wann sich das Portal wirtschaftlich gesehen für die JuFi lohnt, wird im folgenden Abschnitt die Amortisationsdauer berechnet. Hierfür werden Schätzungen der JuFi anhand der aktuellen Arbeitsabläufe genutzt. Die einzelnen Vorgänge und die genauen Angaben zur Zeiteinsparung finden sich in Tabelle 3: Zeiteinsparung wieder. Einsparungen ergeben sich einerseits bei der Bearbeitung von Bestellungen und andererseits bei der Vorbereitung der Hardwareverkäufe. Die JuFi hat nach den Schätzungen durchschnittlich 15 Bestellungen pro Sitzung, welche einmal in der Woche stattfindet. Die ersten 10 Bestellungen haben bei beiden Lösungen die gleiche Bearbeitungszeit, für die restlichen 5 Bestellungen ließe sich mit dem neuen Online Shop die Bearbeitungszeit einsparen. Dadurch würden sich die Kosten für das Personal reduzieren, da 4 Auszubildende der Abteilungen Vertrieb, Finanzbuchhaltung, Informatik und der Geschäftsleitung an dem Bearbeitungsprozess einer Bestellung beteiligt sind. Außerdem würde die Vorbereitung der Hardwareverkäufe nur noch halb so lang dauern, da die IT nur die „Hardwareverkauf Geräte“ – Liste pflegen und keinen GIN Eintrag mit Anmeldeformular erstellen müsste.

Vorgang	Anzahl im Monat	Zeit Vorher	Zeit Nachher	Einsparung pro Monat
Bearbeitung von Bestellungen (händisch)	20	200min	0min	200min
Vorbereitung Hardwareverkauf	0,17 (2 pro Jahr)	120min	60min	10,2min
Einsparung pro Monat gesamt				210,2min

Tabelle 3: Zeiteinsparung

Berechnung der Amortisationsdauer:

$$210,2 \text{ min/Monat} \cdot 12 \text{ Monate/Jahr} = 2.522,4 \text{ min/Jahr} = 42,04 \text{ h/Jahr} \quad (1)$$

$$42,04 \text{ h/Jahr} \cdot (4 \cdot 7,77\text{€})^4 = 1.306,60 \text{ €/Jahr} \quad (2)$$

$$\frac{1.844,98 \text{ €}}{1.306,60 \text{ €/Jahr}} = 1,41 \text{ Jahre} \approx 17 \text{ Monate} \quad (3)$$

⁴ Vier Auszubildende mit einem Stundenlohn von 7,77€ (2.5 Projektkosten)

Das JuFi Portal muss mindestens 17 Monate eingesetzt werden, damit sich die Kosten für die Entwicklung und die Einsparungen ausgleichen. Mit der Grafik A.3: Amortisation auf der Seite ii, soll die Berechnung der Amortisationsdauer noch einmal bildlich veranschaulicht werden.

2.7 Qualitätsanforderungen

Es wurden folgende Anforderungen festgelegt, um die Qualität und Funktionalität des Portals zu gewährleisten:

- Bei der Eingabe einer ungültigen Menge soll der Nutzer ein visuelles Feedback bekommen und gebeten werden seine Eingabe zu korrigieren
- Bei der Artikelübersicht soll zu sehen sein, wenn ein Artikel nicht verfügbar ist
- Pflichtfelder müssen auf Inhalt geprüft werden, wenn ein Pflichtfeld nicht gefüllt ist soll der Nutzer mit einem visuellen Feedback gebeten werden die Felder zu füllen

3 Durchführungsphase

Bevor mit der eigentlichen Durchführungsphase begonnen wurde, wurde zunächst ein Product Backlog im Azure DevOps⁵ angelegt. Dieser Product Backlog enthält alle Anforderungen an das JuFi Portal. Auf Seite iii befindet sich ein Screenshot A.4: JuFi Portal Product Backlog der das Epic, die Features und die Product Backlog Items⁶ zeigt.

3.1 Zielplattform

SharePoint wurde als Zielplattform gewählt, da Microsoft O365 als Collaboration Plattform bei der Geberit verwendet wird. Außerdem realisiert die JuFi die Hardwareverkäufe bereits mit SharePoint, somit müssen sich die IT-Mitarbeiter der JuFi in keine neue Plattform einarbeiten.

3.2 Klassendiagramme

Auf der Seite iii befindet sich das Klassendiagramm A.5: Klassendiagramm JuFi Portal und auf der Seite iv das Klassendiagramm A.6: Klassendiagramm Hardwareverkauf.

⁵ DevOps ermöglicht es zuvor getrennten Rollen wie Entwicklung, IT-Betrieb, Qualitätstechnik und Sicherheit, sich zu koordinieren und zusammenzuarbeiten, um bessere und zuverlässigere Produkte zu liefern. (siehe <https://azure.microsoft.com/de-de/overview/what-is-devops/>)

⁶ Die Begriffe Epic, Feature und Product Backlog Item sind Teil der Vorgehensmodell „Scrum“, welches zur agilen Softwareentwicklung dient (siehe <https://de.wikipedia.org/wiki/Scrum>)

3.3 SharePoint Seite und Microsoft Teams erstellen und einrichten

Die Grundlage des JuFi Portals bildet eine SharePoint Seite und ein Team in Microsoft Teams. Die Seite und das Team wurden zunächst vom O365 Administrator erstellt und dann von mir eingerichtet. Als Vorlage für die Webseite diente die bereits vorhandene Webseite des Drittanbieters. Diese wurde auf der SharePoint Seite inhaltlich nachgebaut. Das Einrichten der SharePoint Seite beinhaltete außerdem das Anlegen der Unterseiten „Online-Shop“ und „Hardwareverkauf“, sowie das Anlegen der Listen „Artikel“, „Bestellte Artikel“, „Bestellungen“, „Hardwareverkauf Anmeldungen“ und „Hardwareverkauf Geräte“. Diese Listen wurden später zum Entwickeln des Online-Shops und des Hardwareverkauf-Formulars benötigt.

Nachdem die SharePoint Seite inklusive Listen angelegt und eingerichtet waren, wurden die für die JuFi-Sitzungen relevanten Listen im Team als Registerkarte hinzugefügt. Das Team und der Reiter ist im Anhang A.7: Geberit Junior Firma Team auf Seite iv abgebildet.

3.4 Online Shop Webpart Programmierung

Da die Entwicklung des Online Shop WebParts der Kernbestandteil des gesamten Projektes darstellt, wird dieser im folgenden Abschnitt genauer erläutert. Um die Entwicklung effektiv und effizient durchführen zu können, ist eine geeignete Entwicklungsumgebung von großer Bedeutung. Dafür wurde VS Code⁷ verwendet.

Zu Beginn der Entwicklung wurde zunächst ein SPFx Projekt angelegt. Um den WebPart während der Entwicklung zu testen wurde eine DEV Seite (eine nicht produktive SharePoint Seite) angelegt, welche als Testumgebung dient. Die DEV Seite ist aufgebaut wie die produktive SP Seite und beinhaltet auch dieselben Listen. Nur die Entwickler haben Zugriff auf die DEV Seite. Auf dieser Seite konnte man den WebPart aufrufen und seine Änderungen sofort anschauen. Um später im Online Shop Artikel anzeigen zu können wurde die „Artikel“-Liste gepflegt. Dazu wurde für jeden Artikel des aktuellen Sortiments ein Eintrag mit Titel, Preis, Farbe, Verfügbarkeit, Beschreibung und Bild erstellt. Mit den Inhalten dieser Liste konnte anschließend eine Artikelübersicht programmiert werden. Dazu wurden in der Datei „Articleoverview.tsx“ alle Einträge der „Artikel“-Liste abgerufen und in „Article.tsx“ ausgegeben. Auf Seite v ist ein Screenshot A.8: Artikelübersicht abgebildet. Nachdem die Artikelübersicht fertig war wurde die Warenkorb-Logik programmiert. Die Warenkorb-Logik beinhaltet das Hinzufügen und Entfernen von Artikeln und die Änderung der Menge. Ein wichtiger Bestandteil der Warenkorb-Logik ist die Überprüfung der Verfügbarkeit eines Artikels sowie die Mindestbestellmenge von einem Stück. Für den Fall, dass ein Artikel nicht verfügbar ist oder

⁷ Visual Studio Code ist ein kostenloser Quelltext-Editor von Microsoft.
(siehe https://de.wikipedia.org/wiki/Visual_Studio_Code)

eine ungültige Menge angegeben wird, wurden zwei Dialoge entwickelt, welche in den Anhängen A.9: Dialog – Artikel nicht verfügbar und A.10: Dialog – ungültige Menge auf Seite v und vi abgebildet sind. Außerdem wird überprüft ob sich etwas im Warenkorb befindet. Mit einem leeren Warenkorb kommt der Kunde nicht weiter zum Bestellabschluss. Ein Screenshot des Warenkorbs ist im Anhang A.11 Screenshot Warenkorb auf Seite vi zu finden. Für den Bestellabschluss wurde eine Maske programmiert, welche die Felder Anrede, Vorname, Nachname, Telefon, E-Mail, Bezahlmethode (Bar oder Kostenstelle) und Anmerkungen beinhaltet. Die Pflichtfelder sind dabei Vorname, Nachname, E-Mail und die Bezahlmethode. Wenn die Bezahlmethode Kostenstelle angeklickt wird, öffnet sich ein weiteres Textfeld für die Eingabe der Kostenstelle. Auf der Seite vii ist die Bestellabschluss-Maske (A.12: Bestellabschluss) abgebildet. Nach Bestellabschluss, welcher mit klicken auf den „Kaufen“ Button ausgelöst wird, werden die Informationen der Bestellung in die „Bestellungen“-Liste und „bestellte Artikel“-Liste geschrieben. Auszüge der Dateien „Articleoverview.tsx“, „WarenkorbOverview.tsx“ und „BestellAbschluss.tsx“ finden sich im Anhang A.13: Screenshots Code auf Seite vii.

3.5 Power Automate Flows

Für die Bestellbestätigung welche der Kunde erhält und die Bestellbenachrichtigung welche an die JuFi zur Bearbeitung gesendet werden soll, wurden Power Automate Flows erstellt. Diese Flows werden ausgeführt sobald die Liste „Bestellungen“ einen neuen Eintrag bekommt. Die Flows ziehen die Informationen der Bestellung aus den Listen „Bestellungen“ und „Bestellte Artikel“ und schreiben sie in einen E-Mail Body, welcher dann an den Kunden und die JuFi versendet wird. Die JuFi kann nun auch für Bestellungen die per Telefon oder persönlich getätigt werden Bestellbestätigungen und Bestellbenachrichtigungen senden. Die Mitarbeiter der JuFi müssen dafür nur die Listen händisch füllen, oder selbst über den Online Shop die Bestellung tätigen. Eine Übersicht über die Schritte die die Flows durchlaufen, ist in dem Anhang A.14: Power Automate Flow auf Seite x abgebildet.

3.6 Hardwareverkauf Webpart Programmierung

Der Hardwareverkauf WebPart besteht aus einer Anmeldemaske mit den Feldern Vorname, Nachname, Personalnummer, Telefon, E-Mail, Gesellschaft und Kategorie (siehe Anhang A.15: Hardwareverkauf Maske auf Seite xi). Da an jedem Hardwareverkauf andere Geräte verkauft werden, füllt sich das „Kategorie“- Drop Down mit den Einträgen der Liste „Hardwareverkauf Geräte“. Die IT-Mitarbeiter der JuFi müssen also vor jedem Hardwareverkauf diese Liste pflegen, die Anmeldungen werden nach dem Klicken auf Speichern in der Liste „Hardwareverkauf Anmeldungen“ gespeichert.

3.7 Testing / Code-Review

Getestet wurde das Portal hauptsächlich während der Entwicklung. Änderungen und Funktionen wurden mit mehreren Prüfungen, unmittelbar nach der Implementierung, getestet. Außerdem wurden nach der Fertigstellung alle WebParts noch einmal mit Use Cases⁸ getestet. Bevor die WebParts auf die Produktivseite geladen wurden, wurde zusammen mit zwei Entwicklern der Collaborations Applications Abteilung ein Code-Review durchgeführt.

3.8 Einrichtung

Für die Einrichtung mussten die WebParts in die Produktivseite implementiert werden. Außerdem musste sichergestellt werden das bestimmte Seiten und Listen nicht von allen Mitarbeitern aufgerufen bzw. bearbeitet werden können. Ein Auszug des Berechtigungskonzept (A. 16: Berechtigungskonzept Listen) für die Listen findet sich im Anhang auf der Seite xi wieder

⁸ Ein Use Case, oder auch Anwendungsfall genannt bündelt alle möglichen Szenarien, die eintreten können, wenn ein Benutzer versucht in einer Software ein bestimmtes Ziel zu erreichen. (siehe <https://de.wikipedia.org/wiki/Anwendungsfall>)

4 Projektabschlussphase

4.1 Abnahme

Nach der Fertigstellung des JuFi Portals konnte das Portal der JuFi zur Endabnahme vorgelegt werden. Aufgrund mehrerer Abstimmungen wurde einem IT-Mitarbeiter der JuFi während der Durchführungsphase öfters die aktuelle Version des Portals präsentiert. In einer JuFi-Sitzung wurde den restlichen JuFi Mitarbeitern und der Ausbildungsleitung in Form einer Schulung das Portal vorgestellt.

4.2 Dokumentation

Die Dokumentation des JuFi Portals setzt sich aus zwei Bestandteilen zusammen: diese Projektdokumentation und einem Benutzerhandbuch für die JuFi. In der Projektdokumentation werden die einzelnen Phasen des Projektes beschrieben, die Dokumentation wurde während der gesamten Laufzeit geschrieben und am Ende des Projektes finalisiert. Das Benutzerhandbuch für die JuFi enthält Informationen über den Aufbau und die Funktionsweise des Portals. Es soll den Mitarbeitern als Anhaltspunkt für Nachfragen zur Verfügung stehen und zur Einarbeitung der neuen JuFi in das Portal dienen. Ein Auszug des Benutzerhandbuchs befindet sich auf Seite xi A. 17: Auszug Benutzerhandbuch.

4.3 Soll-Ist-Vergleich

Bei einer rückblickenden Betrachtung des IHK-Abschlussprojektes kann festgehalten werden, dass die im Abschnitt 2.3 (Projektphasen) erstellte Zeitplanung im Großen und Ganzen eingehalten werden konnte, wie in Tabelle 4: Soll-Ist-Vergleich zu erkennen. Lediglich die Durchführungs- und Implementierungsphase weisen Abweichungen auf, welche für zukünftige Projekte bedacht werden sollten.

Projektphase	Soll	Ist	Differenz
Analysephase	3 h	3 h	0 h
Durchführungsphase	35,5 h	40 h	+ 4,5 h
Implementierungsphase	12,5 h	8h	- 4,5 h
Übergabe	4 h	4 h	0 h
Erstellen der Dokumentation	15 h	15 h	0 h
Gesamt	70 h	70 h	0 h

Tabelle 4: Soll-Ist-Vergleich

4.4 Ausblick

Obwohl die Anforderungen, bis auf die Anbindung an das GIN, realisiert werden konnten, gibt es jetzt schon Ideen bzw. Erweiterungsvorschläge für das Portal. Man könnte beispielsweise einen weiteren WebPart programmieren, der eine Bestellübersicht für die JuFi darstellt und die Möglichkeit bietet, Bestellungen dort zu bearbeiten, Rechnungen zu exportieren und diese dann direkt an die Geberit Finanzbuchhaltung weiterzuleiten. Somit müsste die JuFi Rechnungen nicht mehr von Hand schreiben. Eine Anbindung der „Lager“-Excel der Logistik mit Dashboards zu aktuellen Beständen und eine Benachrichtigung beim Erreichen des Meldebestands wäre auch möglich. Der Bestellübersicht-WebPart und die Dashboards könnte man im Microsoft Teams als Registerkarten hinzufügen, so dass die JuFi für Ihre Arbeit und Sitzungen nur ihr Teams braucht. Den Online-Shop könnte man ausbauen mit einer „Mein Profil“-Übersicht, in der Geberit Mitarbeiter ihre getätigten Bestellungen einsehen können.

4.5 Lessons Learned

Bei der Projektumsetzung konnte ich wertvolle Erfahrungen bzgl. der Arbeit an einem Projekt bzw. dem Projektmanagement sammeln. Sehr wichtig war hierbei auch die stetige Kommunikation untereinander und die Rücksprache mit der JuFi, um eine reibungslose und erfolgreiche Projektumsetzung zu garantieren. Außerdem konnten neue Erkenntnisse in Bezug auf die agile Softwareentwicklung gewonnen werden. Die Realisierung des Projektes konnte nicht nur einen Mehrwert für die JuFi bieten, sondern war auch eine große Bereicherung für mich.

4.6 Fazit

Für das JuFi Portal wurden bis auf die Einbindung in das GIN alle Anforderungen und Wünsche der JuFi umgesetzt. Es ist möglich Bestellungen über den neu programmierten Online-Shop aufzugeben. Zudem können sich die Geberit Mitarbeiter über das JuFi Portal für anstehende Hardwareverkäufe anmelden.

Quellenverzeichnis

SPFx (Framework). Abgerufen 31.05.2021, von

<https://docs.microsoft.com/de-de/sharepoint/dev/spfx/sharepoint-framework-overview>

Azure DevOps (Tool). Abgerufen 31.05.2021 von

<https://azure.microsoft.com/de-de/overview/what-is-devops/>

Epic, Features und Product Backlog Items (Scrum) – Wikipedia. Abgerufen 31.05.2021, von

<https://de.wikipedia.org/wiki/Scrum>

VS Code (IDE) – Wikipedia. Abgerufen 31.05.2021, von

https://de.wikipedia.org/wiki/Visual_Studio_Code

Use Cases – Wikipedia. Abgerufen 31.05.2021, von

<https://de.wikipedia.org/wiki/Anwendungsfall>

A Anhang

A.1 Ausführliche Zeitplanung

Analysephase	3 h
1. Ist-Analyse durchführen	3 h
Durchführungsphase	35,5 h
1. Setup for Development	2 h
2. Erstellen der SharePoint Seite (Webseite)	3 h
3. Erstellen des Microsoft Teams	1,5 h
4. Erstellen der Listen	3 h
5. Realisierung Online Shop	21 h
6. Erstellen der Bestellbenachrichtigung und Bestellbestätigung	3 h
7. Erstellen des Hardwareverkauf WebParts	2 h
Implementierungsphase	12,5 h
1. Testing	8 h
2. Einbindung der Listen in Teams	0,5 h
3. Umzug JuFi Dokumente in Teams	2 h
4. Einbindung JuFi Portal in das Geberit Intranet	2 h
Übergabe	4 h
1. Übergabe vorbereiten	2 h
2. Übergabe an die Geberit Junior Firma	2 h
Erstellen der Dokumentation	15 h
1. Erstellen der Projektdokumentation	15 h

Tabelle 5: Zeitplanung

A.2 Verwendete Ressourcen

Hardware

- Büroarbeitsplatz mit Laptop (Dell LATITUDE E5450)

Software

- Windows 10 Enterprise – Betriebssystem
- Visual Studio Code – Entwicklungsumgebung
- Microsoft SharePoint
- Azure DevOps – Plattform für kollaborative Softwareprojekte
- Microsoft Power Automate – Workflow Automatisierung

Personal

- Entwickler – Umsetzung des Projektes
- IT-Auszubildender der Geberit Junior Firma – Festlegung der Anforderungen
- Softwareentwickler – Betreuung und Review des Codes

A.3 Amortisation

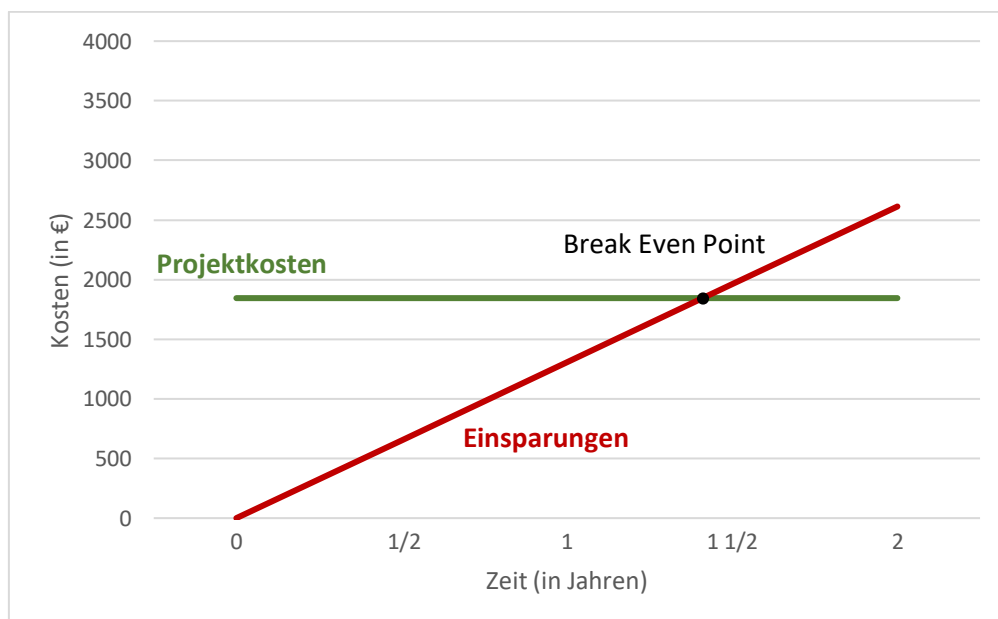


Abbildung 1: Amortisationsdiagramm

A.4 JuFi Portal Product Backlog

15	Epic	▼ JuFi Portal (SPFx)
	Feature	▼ JuFi Portal - Webpart
	Product Backlog...	> JuFi Portal - Webpart - Anzeige Artikelübersicht
	Product Backlog...	> JuFi Portal - Webpart - Setup for Development
	Product Backlog...	> JuFi Portal - Webpart - Listen erstellen
	Product Backlog...	> JuFi Portal - Webpart - Hinzufügen und entfernen von Artikel
	Product Backlog...	> JuFi Portal - Webpart - Bestellbenachrichtigung JuFi
	Product Backlog...	> JuFi Portal - Webpart - Bestellbestätigung User
	Product Backlog...	> JuFi Portal - Webpart - Hardware Verkauf
	Product Backlog...	> JuFi Portal - Webpart - Testing
	Feature	▼ JuFi Portal - Teams & SiteCollection
	Product Backlog...	> JuFi Portal - Teams & SiteCollection - Erstellen der SharePoint Seite
	Product Backlog...	> JuFi Portal - Teams & SiteCollection - Erstellen des Microsoft Teams
	Product Backlog...	> JuFi Portal - Teams & SiteCollection - Einbindung der SharePoint Seite in das Microsoft Teams
	Product Backlog...	> JuFi Portal - Teams & SiteCollection - Umzug JuFi Dokumente
	Feature	▼ JuFi Portal - Project Management
	Product Backlog...	> JuFi Portal - Project Management - Ist-Analyse
	Product Backlog...	> JuFi Portal - Project Management - Projektdokumentation erstellen
	Product Backlog...	> JuFi Portal - Project Management - Übergabe an die JuFi und Ausbildungsleitung
	Product Backlog...	> JuFi Portal - Project Management - GoLive / Einbindung in das Geberit Intranet
	Product Backlog...	> JuFi Portal - Project Management - Soll-Ist-Vergleich

Abbildung 2: Azure DevOps Product Backlog

A.5 Klassendiagramm JuFi Portal

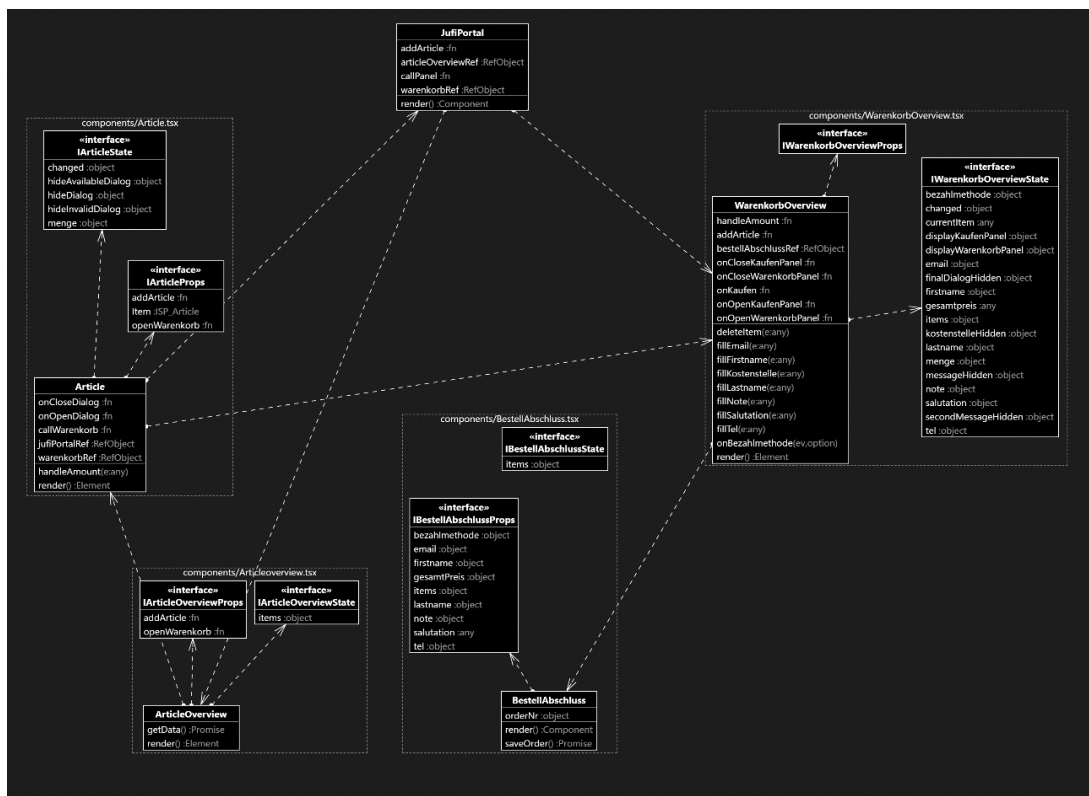


Abbildung 3: JuFi Portal Klassendiagramm

A.6 Klassendiagramm Hardwareverkauf

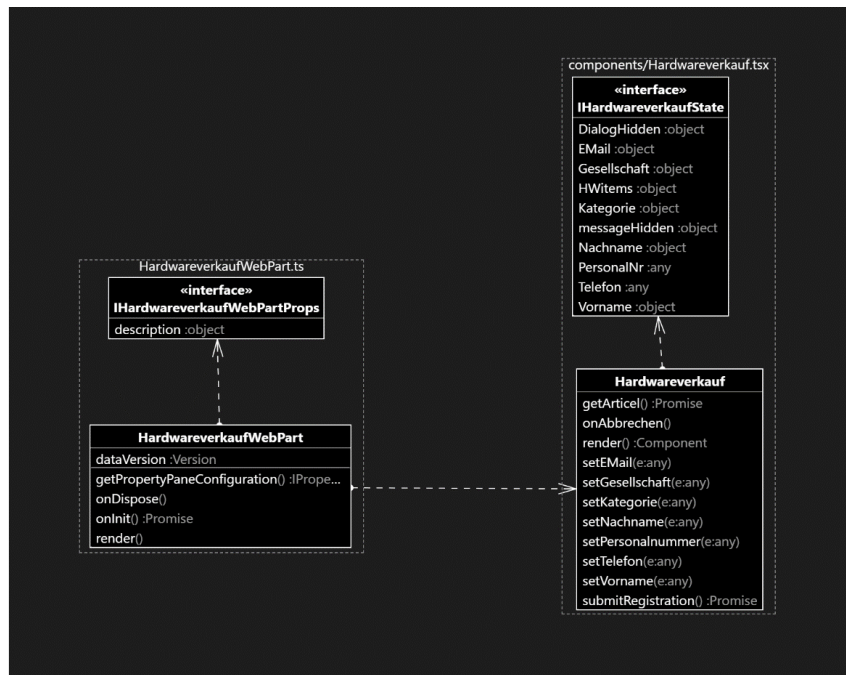


Abbildung 4: Hardwareverkauf Klassendiagramm

A.7 Geberit Junior Firma Team

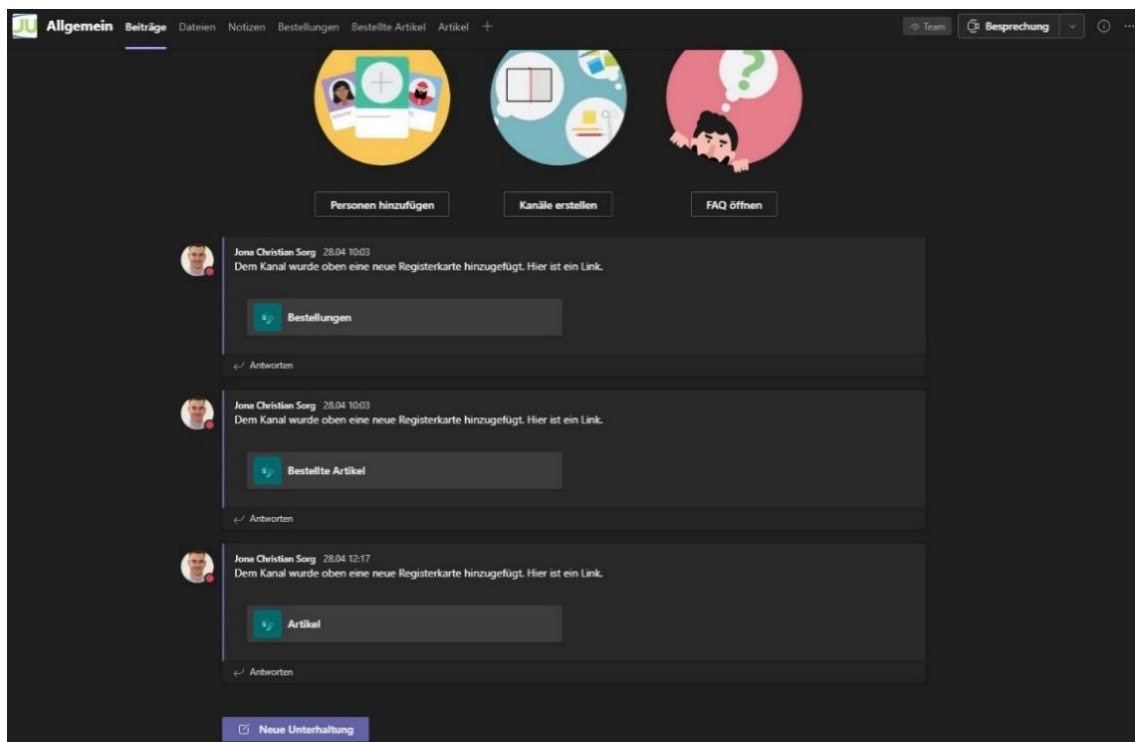


Abbildung 5: JuFi Teams

A.8 Artikelübersicht

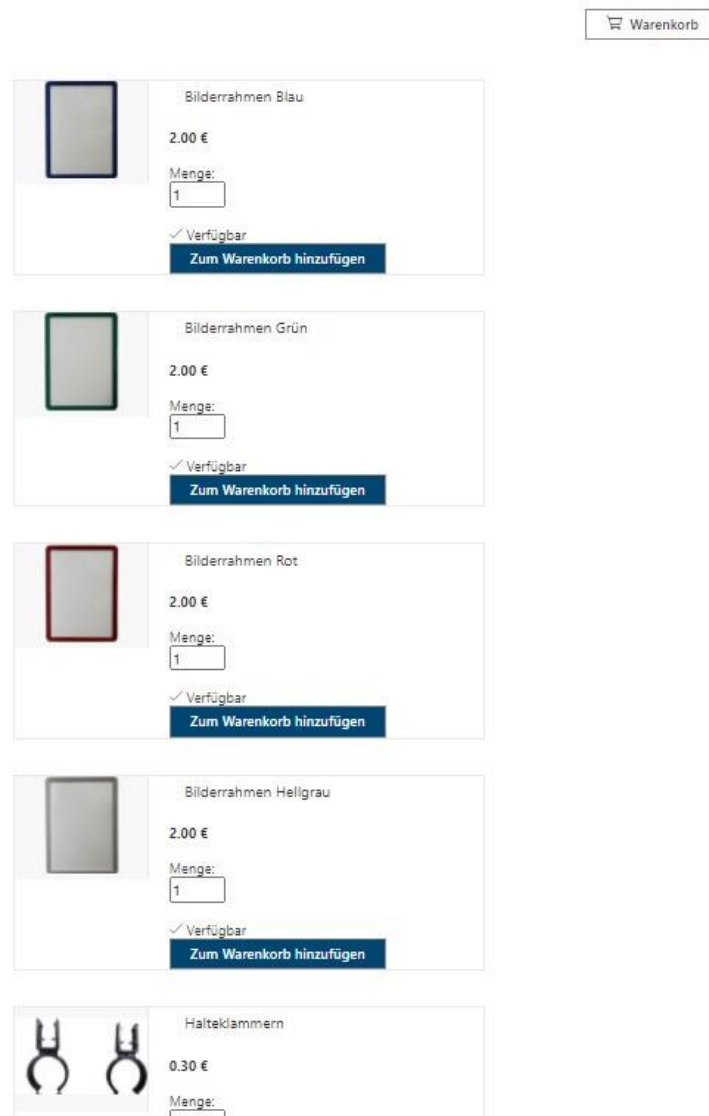


Abbildung 6: Online-Shop Artikelübersicht

A.9: Dialog – Artikel nicht verfügbar



Abbildung 7: Artikel nicht verfügbar Dialog

A.10: Dialog – ungültige Menge



Abbildung 8: Menge ungültig Dialog

A.11 Screenshot Warenkorb

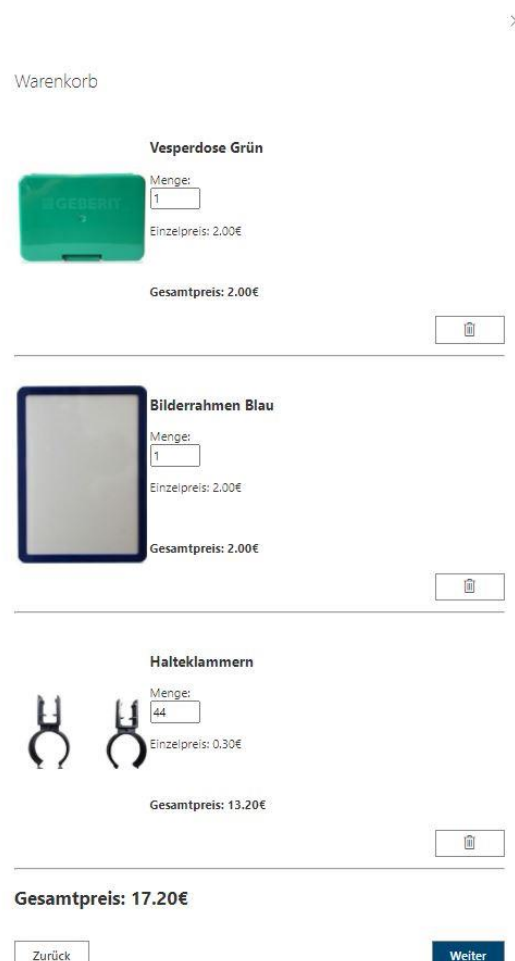


Abbildung 9: Warenkorb

A.12 Bestellabschluss

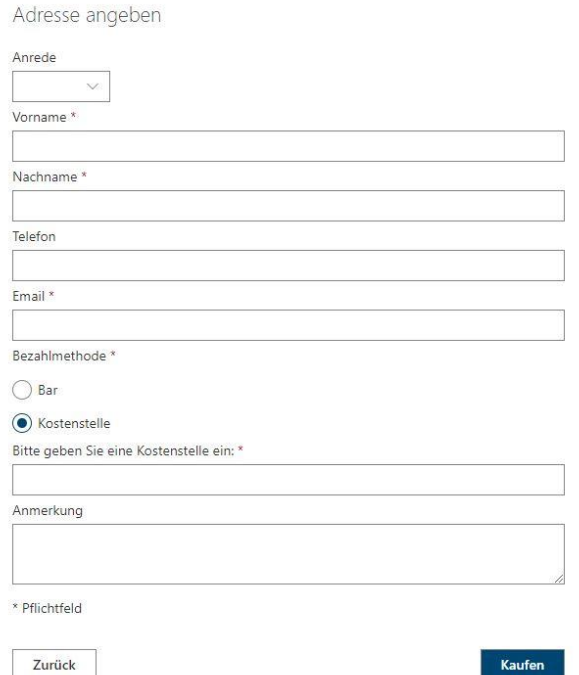


Abbildung 10: Bestellabschluss

A.13 Screenshots Code

```
export default class ArticleOverview extends React.Component<IArticleOverviewProps, IArticleOverviewState> {
  constructor(props) {
    super(props);

    this.getData();

    this.state = {
      items: []
    };
  }

  //Add Article to Overview
  public render() {
    return <div>
      {this.state.items.map(x => {
        return <Article Item={x} openWarenkorb={this.props.openWarenkorb} addArticle={this.props.addArticle}></Article>;
      })}
    </div>;
  }

  //Get Articles from List
  private async getData() {
    const items: ISP_Article[] = await sp.web.lists.getByTitle(_constants.Lists.Articles).items.get();
    console.log(items);
    this.setState({
      items: items
    });
  }
}
```

Abbildung 11: Auszug ArticleOverview.tsx


```

public render() {
  return <div>
    <Panel>
      headerText={"Warenkorb"}
      isOpen={this.state.displayWarenkorbPanel}
      closeButtonAriaLabel="Close"
      type={PanelType.medium}
    >
      {this.state.items.map(x => {
        let imageobj: ISP_Image = null;
        if (x != null) {
          imageobj = JSON.parse(x.Bild);
        }

        if (this.state.changed == true && x.Title == this.state.currentItem) {
          let diff = this.state.menge - x.Menge;
          let newPrice = Number(this.state.gesamtpreis + (diff * x.Preis)).toFixed(2);
          let menge = this.state.menge;
          x.Menge = menge;
          this.setState({
            gesamtpreis: Number(newPrice),
            menge: 0,
            changed: false,
            currentItem: null
          });
        }
      })}

      return <div>
        <br />
        <table>
          <tr>
            <td>
              <img src={` ${imageobj.serverUrl}/${imageobj.serverRelativeUrl}` } alt={imageobj.fieldName} width={150} />
            </td>
            <td>
              <h3>{x.Title}</h3>
              <form>
                <label>Menge:</label> <br />
                <input type="number" defaultValue={' ' + x.Menge} min="1" onChange={this.handleAmount.bind(this, x.Title)} style={{ width: 50, height: 20 }} />
              </form>
              <p>Einzelpreis: {Number(x.Preis).toFixed(2)}€</p>
              <br />
              <h4>Gesamtpreis: {Number(Number(x.Menge) * x.Preis).toFixed(2)}€</h4>
            </td>
          </tr>
        </table>
        <DefaultButton style={{ float: 'right' }} iconProps={DeleteIcon} onClick={() => this.deleteItem(x)} /><br /><br />
      </div>
    </div>
  )}

  <h2>Gesamtpreis: {Number(this.state.gesamtpreis).toFixed(2)}€</h2>
  <br />
  <DefaultButton style={{ float: 'left' }} text="Zurück" onClick={this.onCloseWarenkorbPanel} />
  <PrimaryButton style={{ float: 'right' }} text="Weiter" onClick={this.onOpenKaufenPanel} />
  <br /><br /><br />
  <div hidden={this.state.messageHidden}>
    <MessageBar>
      messageBarType={MessageBarType.error}
      isMultiline={false}
      dismissButtonAriaLabel="Close"
      hidden={this.state.messageHidden}
    >
      Ihr Warenkorb ist leer.
    </MessageBar>
  </div>
</Panel>

```

Abbildung 12: Auszug WarenkorbOverview.tsx

```
export default class BestellAbschluss extends React.Component<IBestellAbschlussProps, {}> {
  public orderNr: string;

  public async saveOrder() {
    let items = this.props.items;
    let salutation;

    if (this.props.salutation.charAt(0) == "H") {
      salutation = "Herr";
    } else if (this.props.salutation.charAt(0) == "F") {
      salutation = "Frau";
    } else if (this.props.salutation.charAt(0) == "D") {
      salutation = "Divers";
    } else {
      salutation = "keine Angabe!";
    }

    const order: ISP_Order[] = await sp.web.lists.getByTitle(_constants.Lists.Orders).items.get();
    let number = order.length + 1;
    let nr = '' + number;

    for (let i = nr.length; i < 5; i++) {
      nr = '0' + nr;
    }

    this.orderNr = nr;

    if (items != []) {
      for (let x of items) {
        await sp.web.lists.getByTitle(_constants.Lists.OrderedArticles).items.add({
          BestellNr: this.orderNr,
          Artikel: x.Title,
          Einzelpreis: x.Preis,
          Anzahl: x.Menge,
        });
      }
    }

    const orders: IItemAddResult = await sp.web.lists.getByTitle(_constants.Lists.Orders).items.add({
      Title: this.orderNr,
      Anrede: salutation,
      Vorname: this.props.firstname,
      Nachname: this.props.lastname,
      Telefon: Number(this.props.tel),
      Email: this.props.email,
      Anmerkung: this.props.note,
      Bezahlmethode: this.props.bezahlungsmethode,
      Status: "Neu",
    });
  } else {
    console.log("Es wurden keine Items übergeben");
  }
}
```

Abbildung 13: Auszug BestellAbschluss.tsx

A.14 Power Automate Flow

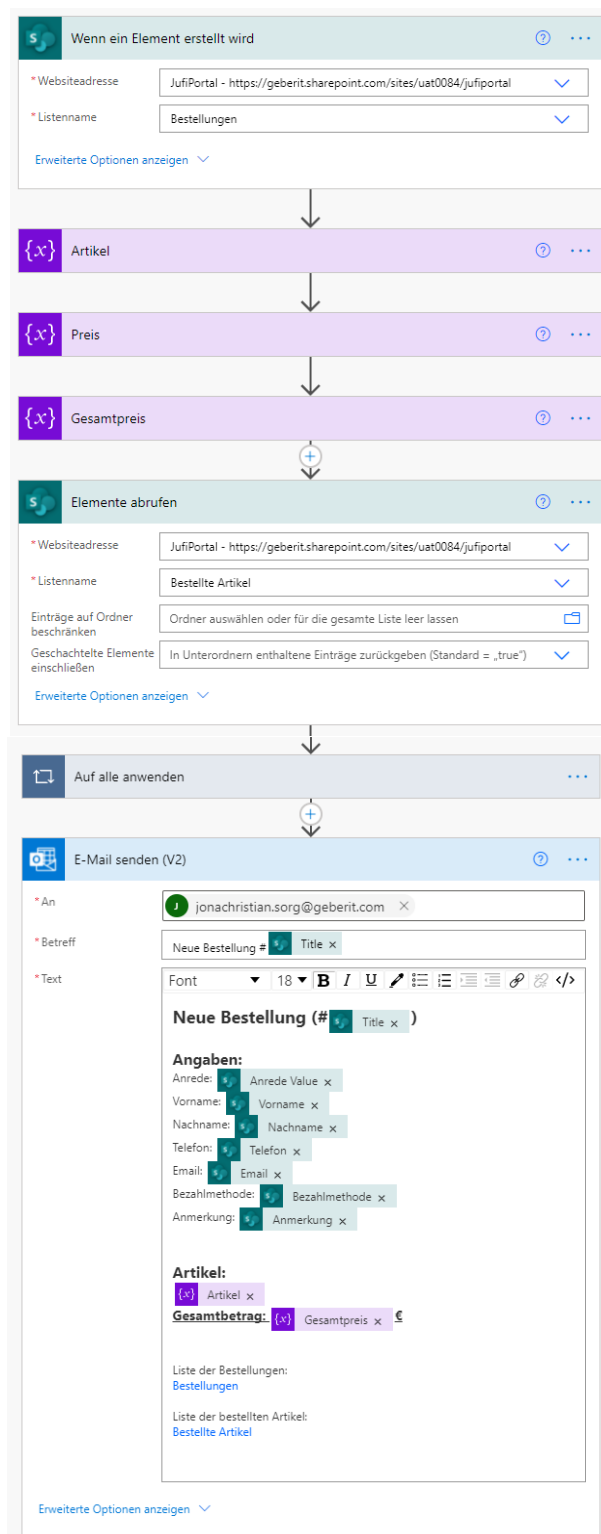


Abbildung 14: Power Automate Flow

A.15 Hardwareverkauf Maske

Hardwareverkauf Anmeldung

Vorname *

Nachname *

Personalnummer *

Telefon

E-Mail *

Gesellschaft *

Kategorie *

Abbildung 15: Hardwareverkauf Anmeldemaske

A.16 Berechtigungskonzept Listen

Vollzugriff: Entwickler, IT-Mitarbeiter der JuFi

Bearbeiten: JuFi User, Mitarbeiter der JuFi

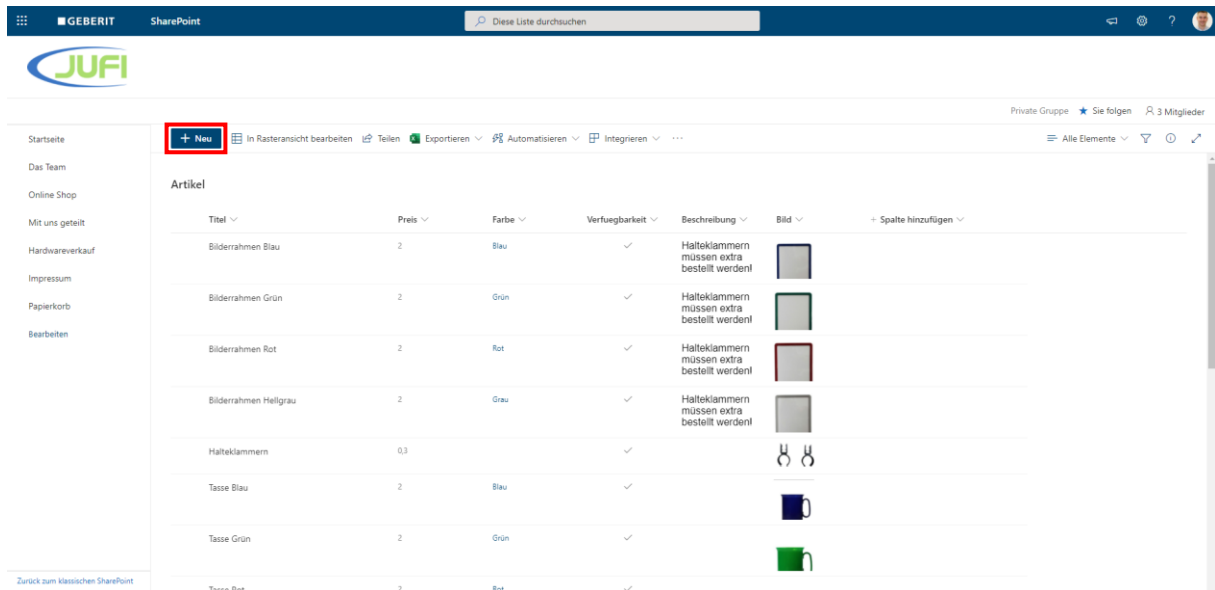
Lesen: Geberit Mitarbeiter

A.17 Auszug Benutzerhandbuch








Der folgende Auszug aus dem Benutzerhandbuch erläutert Schritt für Schritt, wie ein neuer JuFi Artikel zum Online-Shop hinzugefügt werden kann.

Artikel hinzufügen

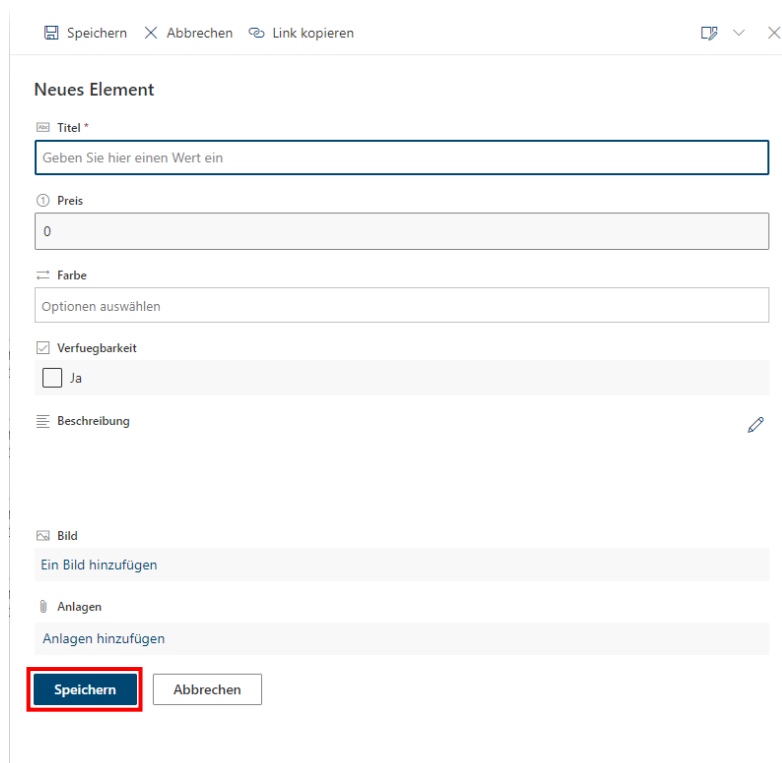
Um einen neuen Artikel im Sortiment aufzunehmen, muss er als Eintrag in der Liste „Artikel“ ([Geberit Junior Firma - Artikel - Alle Elemente \(sharepoint.com\)](#)) hinzugefügt werden. Einen neuen Eintrag kann man mit dem „Neu“-Button erstellen.



The screenshot shows the JuFi Portal SharePoint interface. The top navigation bar includes the Geberit logo and the text 'JuFi Portal'. The main content area displays a list of articles with columns: Titel, Preis, Farbe, Verfügbarekeit, Beschreibung, and Bild. The 'Neu' button is highlighted in red.

Titel	Preis	Farbe	Verfügbarekeit	Beschreibung	Bild
Bilderrahmen Blau	2	Blau	✓	Halteklammern müssen extra bestellt werden!	
Bilderrahmen Grün	2	Grün	✓	Halteklammern müssen extra bestellt werden!	
Bilderrahmen Rot	2	Rot	✓	Halteklammern müssen extra bestellt werden!	
Bilderrahmen Hellgrau	2	Grau	✓	Halteklammern müssen extra bestellt werden!	
Halteklammern	0,3		✓		
Tasse Blau	2	Blau	✓		
Tasse Grün	2	Grün	✓		
Tasse Rot	2	Rot	✓		

Nach dem Klicken öffnet sich ein Panel.



The screenshot shows the 'Neues Element' form. The form includes fields for Titel, Preis, Farbe, Verfügbarekeit, Beschreibung, and Bild. The 'Speichern' button is highlighted in red.

Neues Element

Titel *
Geben Sie hier einen Wert ein

Preis
0

Farbe
Optionen auswählen

Verfügbarekeit
☐ Ja

Beschreibung

Bild
Ein Bild hinzufügen

Anlagen
Anlagen hinzufügen

Speichern **Abbrechen**

Dort müssen alle wichtigen Informationen eingetragen werden. Danach kann der Artikel mit dem „Speichern“-Button gespeichert werden. Der Artikel wird automatisch im Online Shop angezeigt.

Industrie- und Handelskammer
Bodensee – Oberschwaben

AUSBILDUNG

Bestätigung über die durchgeführte Projektarbeit

Ausbildungsberuf:	Fachinformatiker Fachrichtung Anwendungsentwicklung
Vorname, Nachname (Prüfling):	Jøna Sorg
Ausbildungsbetrieb/Praktikumsbetrieb:	Geberit Verwaltungs GmbH
Umschulungsträger:	
Abschlussprüfung:	Sommer 2021
Projektbezeichnung:	JuFi Portal
Projektbeginn:	06.04.2021
Projektfertigstellung:	21.04.2021
Zeitaufwand in Stunden:	70 Stunden

Bestätigung des Ausbildungsbetriebes/Praktikumsbetriebes

Wir bestätigen, dass der/die Auszubildende das oben bezeichnete Projekt einschließlich der Dokumentation im Zeitraum vom 06.04.2021 bis 21.04.2021 selbstständig ausgeführt hat.

Projektverantwortliche/r der Firma:

Vorname, Nachname: Daniel Schramm
Telefon: 07552/934284

Pfullendorf, den 01.06.2021

Ort, Datum


Unterschrift des Projektverantwortlichen

Diese Bestätigung ist der Projektdokumentation beizulegen.

Zeitaufwand für die durchgeführte Projektarbeit

Wurde das Projekt in der kalkulierten Zeit fertig gestellt (einschließlich eventueller Nacharbeit)?

☒ Ja

☐ Nein,

um _____ Stunden früher fertig geworden

um _____ Stunden länger gebraucht

Ausführung

Wurde das Projekt entsprechend dem eingereichten Konzept ausgeführt?

☐ Ja

☒ Nein

Wenn Nein, welche Änderungen ergaben sich?

Die Gliederung weicht von der Grobgliederung im Projektantrag ab. Außerdem wurde das Portal noch nicht in das Geberit Intranet implementiert.

Wurde das Projekt selbstständig und ohne fremde Hilfe ausgeführt?

☒ Ja

☐ Nein

Wenn Nein, Begründung und Umfang der Hilfestellung:

Konnte das Projekt ohne Nacharbeit in einem einwandfreien Zustand übergeben werden?

☒ Ja

☐ Nein

Wenn Nein, Begründung und Umfang der Nacharbeit:

Dokumentation

Wurde die Dokumentation vom Prüfungsteilnehmer selbstständig und ohne fremde Hilfe erstellt?

- ☒ Ja
☐ Nein

Wenn Nein, welche Hilfestellung wurde gegeben:

Entspricht die Dokumentation den betrieblichen Anforderungen?

- ☒ Ja
☐ Nein

Wenn Nein, worin bestehen die Abweichungen:

Eidesstattliche Erklärung

Ich versichere durch meine Unterschrift, dass ich die betriebliche Projektarbeit und die dazugehörige Dokumentation selbstständig in der angegebenen Zeit erarbeitet habe. Alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, wurden von mir als solche kenntlich gemacht. Ebenso bestätige ich, dass ich bei der Erstellung der Dokumentation meiner Projektarbeit weder teilweise noch vollständig Passagen aus Projektarbeiten übernommen habe, die bei der prüfenden oder einer anderen Kammer eingereicht wurden.

Pfullendorf, den 01.06.2021

Ort, Datum


Unterschrift des Prüfungsteilnehmers