

ShopCorner

Gruppe: QSE_8

Gruppenmitglieder:

- Christian Ammann
11713963
- Hafiz Khan
01125098
- Anja Kurzmann
11905151
- Giancarlo Buenaflor
51837398
- Michael Steinkellner
11705362
- Simon Roth
11907576

1. Projektbezeichnung und Entwicklerteam

Id	Name	Rolle	Stellvertretende Rolle
ca	Christian Ammann	Technische_r Architekt_in	Dokumentationsbeauftragter
hk	Hafiz Khan	Teamkoordinator_in	Testleiter_in
ak	Anja Kurzmann	Testleiter_in	Teamkoordinator_in
gb	Giancarlo Buenaflor	UI_UX Leiter_in	Gitmanager
ms	Michael Steinkellner	Dokumentationsbeauftragter	UI_UX Leiter_in,
sr	Simon Roth	Gitmanager	Technische_r Architekt_in

2. Ausgangssituation

Firma A möchte wegen den derzeitigen Umständen einen zusätzlichen Onlineshop erstellen, in dem all ihre Produkte und Angebote dargestellt werden. Die Firma will die Rechnungen des Online-Geschäfts und die Rechnungen des Geschäftslokals nicht mit separaten Systemen erstellen und verwalten; deshalb wollen sie eine einheitliche Applikation, in der die Rechnungen für den Onlinehandel automatisch an den Kunden versandt werden und für den normalen Handel ausgedruckt werden können. Kunden sollten in der Lage sein die angebotenen Produkte zu bestellen. Unter anderem soll die Bestellbestätigung automatisch versandt werden und die Bestellungen für die Firma aufgelistet werden.

3. Projektbeschreibung

Es wird eine Applikation entwickelt, mit der eine Firma ihre Produkte verwalten und online verkaufen kann. Des Weiteren sollen die Rechnungen für den normalen Handel erstellt werden und im Onlinehandel automatisch an den Kunden versandt werden.

Für die Firma soll eine Kaufstatistik (Woche, Monat, Jahr) erstellt werden, um beliebte Produkte zu filtern.

Angebote und Newsletter sollen von der Firma erstellt und auf der Webseite angezeigt werden. Die Newsletter werden, wenn der Kunde sich dafür anmeldet, automatisch mit den derzeitigen Angeboten an den Kunden per E-Mail gesendet. Für die Firma muss im Rahmen der Verwaltung des Onlinehandels eine Liste mit noch nicht versendeten Produkte angezeigt werden.

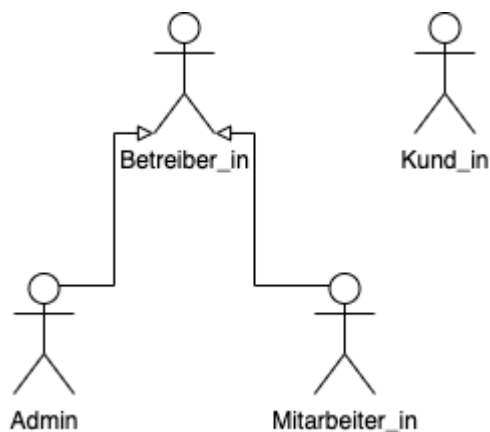
Die Daten zu Rechnungserstellung und Produktverwaltung sowie jene für den Online-Store werden in einer H2 Datenbank gespeichert, auf einem Webserver verarbeitet und über die REST-Schnittstelle dem Client zur Verfügung gestellt. Die

genauen Funktionalitäten der Applikation sind unter “Featureliste”(5.1) beschrieben und aufgelistet.

Wichtig ist, dass die Implementierung und die Verwaltung der Authentifizierung sowie der Berechtigungslevel gut umgesetzt werden, um mögliche Angriffe zu verhindern.

4. Zielgruppen

Die Zielgruppe besteht aus österreichischen Kleinunternehmen, welche vom physischen in den Onlinehandel expandieren wollen. Dabei handelt es sich konkret um von der Umsatzsteuer verpflichtete oder befreite Unternehmen, welche nur auf nationaler Ebene Geschäfte tätigen. Da die Umsatzsteuer von dem jeweiligen Produkt und Unternehmen abhängig ist, kann diese variable eingegeben bzw. exkludiert werden.



Akteur	Rechte im System	Anmerkungen
Betreiber_in		Abstrakter Grundstein für Admin und Mitarbeiter_in
Admin	Kann Aktionen erstellen, Produkte anpassen und Betreiber_innen Konten verwalten.	
Mitarbeiter_in	Rechte werden von Admins festgelegt.	Wird von Admin erstellt
Kund_in	Kann Produkte dem Warenkorb hinzufügen. Auschecken. Vom Newsletter an- und abmelden.	Meldet sich an. Bei Registrierung wird wegen Newsletter gefragt

5. Funktionale Anforderungen und Anwendungsfälle

5.1. Featureliste

Die Priorität ist deklariert als "N" für niedrig, "M" für mittel und "H" für hoch. Durch Schätzung und einer Skala von 1 bis 9 wurde der Zeitaufwand angegeben, wobei 1 für den geringsten und 9 für den höchsten Arbeitsaufwand steht.

Id	Name	Beschreibung	Kunden priorität	Zeitauf wand
1	Betreiberkonten	Es soll initial ein permanent im System erfasstes Adminkonto geben, um andere Betreiber_innenkonten hinzufügen und verwalten zu können. Für Konten gibt es drei Berechtigungsstufen. Kund_innen haben keine besonderen Rechte, Mitarbeiter_innen und Admins haben in den weiteren Features erläuterte Extrarechte. Admins besitzen weiters die Möglichkeit, Mitarbeiter_innen- und Adminkonten anzulegen. Betreiber_innen des Shops können sich über die Shopfront registrieren und ihre Profildaten bearbeiten, sowie das gesamte Konto unwiderruflich löschen.	H	5
2	Authentifizierung	Betreiber_innen können sich mit einem Betreibernamen und einem Passwort anmelden. Nach fünf falschen Einlogversuchen wird ein Konto gesperrt. Diese Sperre kann von Admins wieder aufgehoben werden.	H	5
3	Konto Administration	Admins haben die Möglichkeit andere Betreiber_innenkonten auf niedrigerer Berechtigungsebene zu sperren. Admins können sich alle gesperrten Konten anzeigen lassen und diese auch wieder freischalten. Admins besitzen die Möglichkeit, Mitarbeiter_innenkonten zu Adminkonten aufzustufen. Es besteht ebenso die Möglichkeit, Passwörter wiederherzustellen.	H	4
4	Rechnungen verwalten	Rechnungen werden automatisch bei Bestellungen generiert und gespeichert. Sie sind auch manuell verwaltbar. Mitarbeiter_innen und Admins besitzen auch die Möglichkeit, Rechnungen für vom Webshop entkoppelte (physische) Transaktionen manuell zu erstellen. Es besteht auch die Möglichkeit, Stornorechnungen zu erstellen. Die Umsatzsteuer wird bei der Rechnungserstellung, insofern diese relevant ist, beachtet..	H	7
5	Rechnungen versenden	Rechnungen werden an Kund_innen automatisch per E-Mail versandt nachdem bestellt wurde.	H	3

6	Rechnungen drucken	Rechnungen sollen auch physisch druckbar sein.	N	3
7	Belege	Belege sollen von Betreiber_innen erstellt werden können und immer zu einer ausgehenden Rechnung gehören.	M	4
8	Produkte verwalten	Produkte sollen vom Admin erstellt werden können. Produktgruppe wird ausgewählt um Steuersatz zu bestimmen. Die erstellten Produkte können anschließend im Webshop von Kund_innen erstanden werden.	M	4
9	Statistiken	Für die Rechnungen sind im Frontend mehrere Statistiken grafisch darstellbar. Diese Graphen bieten auch Interaktion. Diese sollen nur für die Betreiber sichtbar sein.	M	7
10	Aktionen	Admins können Aktionen erstellen, welche Kund_innen Rabatte auf selektierte Bestellungen geben. Eine Aktion hat ein Ablaufdatum und einen (oder mehrere) Code zum einlösen.	N	5
11	Newsletter	Im Newsletter werden an Kund_innen wöchentlich die aktuellen Aktionen per E-Mail versandt. Von diesem Newsletter kann der Kunde / die Kundin sich auch abmelden.	N	4
12	Shopfront	Ein Admin kann den Titel des Shops ändern und kleine Designänderungen vornehmen.	H	
13	Produkt kaufen	Kund_innen können Produkte zum Warenkorb hinzufügen und Gutscheincodes einlösen. Kund_innen müssen authentifiziert/eingeloggt sein, um Produkte kaufen zu können.	H	
14	News Page	Zeigt aktuelle Shopneuigkeiten und Aktionen an für Kund_innen.	N	
15	Bestseller	Die meistgekauften Produkte werden in einer eigenen Seite hervorgehoben.	M	
16	Produktsuche/-filter	Kund_innen können Produkte nach Begriffen suchen oder nach Kategorien filtern..	H	
17	Bestellübersicht	Kund_innen können ihre Bestellhistorie einsehen.. Admins können gesamte Bestellhistorie(alle von Kund_innen bestellte Produkte) sehen.	M	
18	Verkaufsbenachrichtigungen	Wenn erwünscht, erhalten Admins Nachrichten per E-Mail für Produktverkäufe.	H	
19	Kauf stornieren	Käufe können innerhalb von einem von Admins angegebenen Zeitraums storniert werden. Produkte können auch als nicht stornierbar deklariert werden.	M	

5.2. Anwendungsfallübersicht

1. Kontoverwaltung (H): Unbedingt erforderlich, kritisch
2. Authentifizierung (H): Unbedingt erforderlich, kritisch
3. Rechnungsverwaltung (H): Unbedingt erforderlich, kritisch
4. Bestellbestätigung (H): Unbedingt erforderlich, kritisch
5. Statistiken anzeigen (M): Grundfunktionalität erforderlich (anzeigen lassen, noch nicht interaktiv)
6. Aktionen (M): Grundfunktionalität erforderlich (können erstellt werden, Codes können beim Kauf angewendet werden)
7. Newsletter (N): Erweiternde Features, können auch nach dem Projekt implementiert werden
8. Produkte/Layout verwalten (H): Unbedingt erforderlich, kritisch
9. Kundenverwaltung(M): Grundfunktionalität erforderlich (anzeigen lassen, noch nicht interaktiv)
10. Einkaufen(H): Unbedingt erforderlich, kritisch
11. Kauf stornieren(M): Grundfunktionalität erforderlich (Kauf stornieren) erweiternd (Betreiber stellt individuell Stornierungszeitraum ein)
12. Bestellungen verwalten(M): Grundfunktionalität erforderlich (Kunde erhält Überblick)

5.3. Iceberglist

Id	Feature, Akteur	Anwendungsfälle	Kundenpriorität	Aufwand	Version	Zuständige Personen
1.1	Kontoverwaltung, Admin	Admin Konto registrieren	H	4	1	ak
1.2	Kontoverwaltung, Admin	Mitarbeiter_in Konto registrieren	H	3	1	sr
1.3	Kontoverwaltung, Admin	Mitarbeiter_in Konto sperren	H	1	1	ak
1.4	Kontoverwaltung, Admin	Mitarbeiter_in Konto freischalten	H	1	1	ak

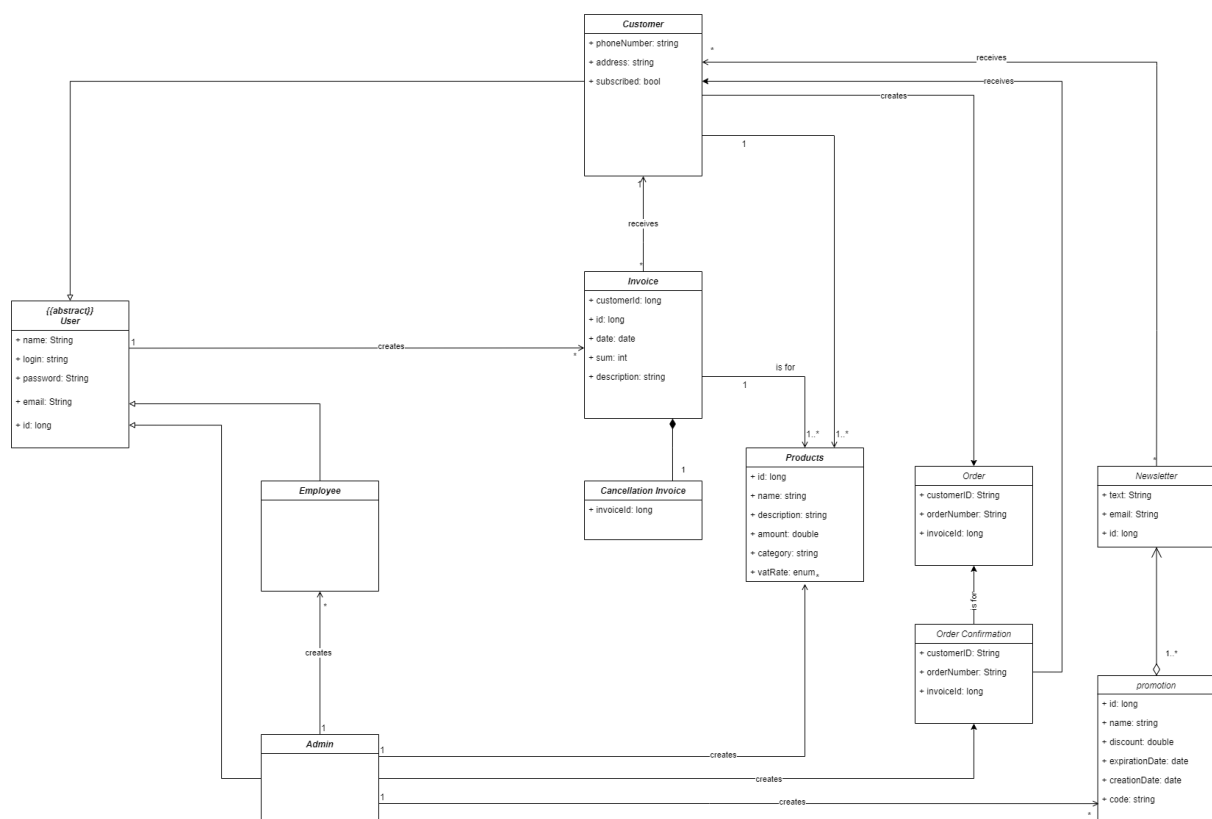
1.5	Kontoverwaltung, Admin	Mitarbeiter_in Konto löschen	H	1	1	ms
1.6	Kontoverwaltung, Admin	Mitarbeiter_in Konto Berechtigungslevel ändern	H	1	1	ms
1.7	Kontoverwaltung, Mitarbeiter_in	Mitarbeiter_in Konto Daten editieren	H	2	1	sr
1.8	Kontoverwaltung, Admin	Admin Konto Daten editieren	H	2	1	sr
1.9	Kontoverwaltung, Admin	Kontenübersicht	H	2	1	ms
1.10	Kontoverwaltung, Kund_in	Kundenkonto registrieren	H	2	1	gb
1.11	Kontoverwaltung, Admin	Kundenkonto sperren	H	1	1	gb
2.1	Authentifizierung, Betreiber	Mit Benutzername und Passwort einloggen	H	4	1	gb
2.2	Authentifizierung, Betreiber	Bei 5 Fehlversuchen Konto sperren	H	2	1	gb
2.3	Authentifizierung, Kund_in	Mit Benutzername und Passwort einloggen	H	2	1	sr
2.4	Authentifizierung, Kund_in	Bei 5 Fehlversuchen Konto sperren	H	1	1	ak
8.1	Produkte/Layout verwalten, Admin	Produkt erstellen	H	6	1	hk, ca
8.7	Produkte/Layout verwalten, Admin	Produkt zur Shopfront hinzufügen	H	4	1	ca
3.1	Rechnungsverwalt ung, Betreiber	Rechnung erstellen	H	7	2	ca
3.4	Rechnungsverwalt ung, Betreiber	Rechnungsübersicht	H	2	2	sr
8.5	Produkte/Layout verwalten, Admin	Produkt löschen	H	2	2	sr

8.6	Produkte/Layout verwalten, Betreiber	Produkt außerhalb der Shopfront detailliert anzeigen	H	2	2	sr
9.1	Kunden- verwaltung, Betreiber	Kunden anzeigen	H	1	2	ms
9.2	Kunden- verwaltung, Betreiber	Kunden löschen	M	1	2	ms
4.1	Bestellbestätigung, Kunde	Bestellbestätigung als E-Mail erhalten. Rechnung als attachement	H	6	2	ms
8.2	Produkte/Layout verwalten, Admin	Produkt bearbeiten	H	4	2	ak
6.1	Aktionen, Admin	Aktion erstellen	M	5	2	sr,hk
6.2	Aktionen, Betreiber	Aktionen anzeigen	M	2	2	ak
10.1	Einkaufen, Kund_in	Kund_in kann Produkte dem Warenkorb hinzufügen	H	7	2	ca, hk
10.2	Einkaufen, Kund_in	Kund_in kann vom Warenkorb aus bestellen	H	5	2	gb
10.3	Einkaufen, Kund_in	Kund_in kann Produkte aus dem Warenkorb entfernen	H	3	2	gb
10.5	Einkaufen, Kund_in	Produkte filtern oder danach suchen	M	5	2	ak
11.1	Kauf stornieren, Kund_in	Kund_in kann einen Kauf innerhalb einer gewissen Zeit stornieren	H	4	2	gb
11.2	Kauf stornieren, Admin	Admin kann Zeitraum für Stornierungen anpassen	M	3	2	ms
3.2	Rechnungsverwalt ung, Betreiber	Rechnung stornieren	H	5	3	hk
3.3	Rechnungsverwalt ung, Betreiber	Stornorechnung erstellen	H	5	3	hk
3.5	Rechnungsverwalt ung, Betreiber	Rechnung drucken	N	3	3	sr

5.1	Statistiken anzeigen, Betreiber	Gewählte Statistik anzeigen lassen	M	4	3	ms
5.2	Statistiken anzeigen, Betreiber	Statistiken filtern nach Kunden	M	3	3	ms
5.3	Statistiken anzeigen, Betreiber	Statistiken filtern nach Produkten	M	2	3	ca
5.4	Statistiken anzeigen, Betreiber	Mit Graphen interagieren	N	4	3	ms
5.5	Statistiken anzeigen, Betreiber	Statistiken als PDF exportieren	N	2	3	ak
7.1	Newsletter, Server	Newsletter erstellen	N	3	3	gb
7.2	Newsletter, Kund_in	Newsletter erhalten	N	2	3	gb
7.3	Newsletter, Kunde	Vom Newsletter abmelden	N	2	3	gb
7.4	Newsletter, Kunde	Zum Newsletter anmelden	N	2	3	ca
8.3	Produkte/Layout verwalten, Admin	Produkt als auslaufend markieren	M	3	3	ak
8.4	Produkte/Layout verwalten, Admin	Produkt sperren	M	2	3	ak
8.7	Produkte/Layout verwalten, Admin	Logo zu Shop hinzufügen	M	1	3	gb
8.8	Produkte/Layout verwalten, Admin	Shop Banner bearbeiten	N	2	3	gb
10.4	Einkaufen, Kund_in	Aktionen können an Bestellung angewendet werden	M	7	3	sr,ca
10.6	Einkaufen, Kund_in	Kund_in kann mit Paypal zahlen	N	6	3	gb,sr

12.1	Bestellungen verwalten, Kund_in	Kund_in kann sich Bestellhistorie ansehen	M	2	3	hk
12.2	Bestellungen verwalten, Kund_in	Kund_in kann von Bestellhistorie direkt stornieren	M	2	3	hk
12.3	Bestellungen verwalten, Betreiber_in	Betreiber_in kann gesamte Bestellhistorie aller Kunden überblicken	N	2	3	hk
12.4	Bestellungen verwalten, Kund_in	Kund_in kann von Bestellhistorie Rechnung als PDF downloaden	N	3	3	ak

6. Domänenmodell



7. Arbeitsstruktur & Grober Projektplan

7.1. Rollenverteilung

Id	Name	Rolle	Stellvertretende Rolle
ca	Christian Ammann	Technische_r Architekt_in	Dokumentationsbeauftragter
hk	Hafiz Khan	Teamkoordinator_in	
ak	Anja Kurzmann	Testleiter_in	Teamkoordinator_in
gb	Giancarlo Buenaflor	UI_UX Leiter_in	Gitmanager
ms	Michael Steinkellner	Dokumentationsbeauftragter	UI_UX Leiter_in,
sr	Simon Roth	Gitmanager	Technische_r Architekt_in

7.2. Horizontale Verantwortlichkeiten

• Technische Architekten: ca, sr

Technische Architekten verwalten die Programm Infrastruktur des Projekts, z.B. Ordnerstruktur und Abhängige JAR Bibliotheken der Software (Dependencies). Zum Expertenwissen der Technischen Architekten zählen eindeutig sehr gute Kenntnisse der verwendeten Programmiersprachen, Toolkits und Frameworks, sowie Architekturelles Design und Software Patterns. Spezifische Horizontale Verantwortlichkeiten für dieses Projekt sind:

- Project Object Model (Maven pom.xml)
 - Dependency Management, vor allem Versionen
 - Plugins
 - verwendete externe Maven Repositories
- Expertenwissen zu allen verwendeten Technologien der Technischen Entwicklung
- Erstellung von Kodierungs-Richtlinien (Checkstyle).
- Erstellung von Richtlinien zur Sourcecode-Dokumentation (Javadoc).
- Design der Programmarchitektur und -komponenten
 - UML Komponentendiagramm auf Interface-Ebene
 - Verteilungsdiagramm

• Teamkoordinatoren: hk, ak

Team Koordinatoren müssen gute Projektmanagement Kenntnisse haben und sind für die Aktualität der Artefakte des laufenden Projektmanagement (z.B. Projektplan) verantwortlich.

- Organisation und Planung (laufende Dokumentation)
 - Projektplan, WBS, siehe Grober Projektplan
 - Risikoanalyse, Iceberglis, Burn-Down Charts
- Controlling & Tracking
 - Stundenlisten, Organisatorische-Tickets vom Tracker

- Statuserhebung für den Statusbericht (Reviews), siehe Lieferkomponenten
 - Kontrolle der Aufgabenverteilung (Arbeit/Developer), siehe Grober Projektplan und Iceberglis
 - Primärer Ansprechpartner für die Auftraggeber
 - Organisation interner und externer Meetings
 - Gruppentreffen (Jour-Fixe), Tutortreffen, IR's, MR's
 - siehe Informationswesen
- **Testleiter_in: ak, hk**
 - Test Infrastruktur
 - Test Bibliotheken, Testdaten (zweite DB).
 - Test Suites: Integration mit Spring Framework um Testdaten zu managen.
 - Sicherstellen, dass Testcode von Produktionscode getrennt wird.
 - Erstellung des Testplans
 - Testvorgehensweise (Codegerüst), Planung von Test-Runs
 - Auswirkungen bei Fehlern, Weitergabeverhalten bei Fehlern (Exceptionhandling)
 - siehe Meilenstein 2 in den Lieferkomponenten
 - Überprüfung der Einhaltung von Testrichtlinien
 - Verwaltung aller Trouble-Tickets im Tracker
 - Überwachung von Integrations- und Systemtests (starke Zusammenarbeit mit dem Technischen Architekten)
 - Regelmäßige Überprüfung aller Unit-Test
- **Gitmanager: sr, gb**
 - Git Workflow
 - Plan festlegen
 - Sicherstellen, dass verschieden branches für Features, Development und Master verwendet werden
 - Sicherstellen, dass commit messages mit Issuenummern versehen sind
- **UI_UX Leiter: gb, ms**
 - Designentwicklung
 - Entwicklung von Mock-Ups
 - Usability gewährleisten
 - Kontrolle über die einheitliche Gestaltung der Website
- **Dokumentationsbeauftragte: ms, ca**
 - Verfügbarkeit der Dokumentation in Gitlab
 - Java API Dokumentation
 - Sicherstellen, dass alle Klassen, Variablen und Methoden in English dokumentiert werden
 - Erstellung von Dokumentationsrichtlinien (Format- und Formatierungsrichtlinien, Spezifikation der Code Conventions, Erstellung von Vorlagen, ...)

- Überprüfung der Einhaltung von Dokumentationsrichtlinien
- Überprüfung der Vollständigkeit von Dokumenten
- Organisation und Archivierung der Dokumente in Gitlab Wiki

7.3. Grober Projektplan

7.3.1. Work Breakdown Structure

Nr.	Arbeitspakete	Anfang	Ende	Persone n- tage	Verant- wortlich
<i>MS.0</i>	<i>Kick-Off</i>	<i>14.04.2021</i>	<i>14.04.2021</i>	<i>1</i>	
1.1	Anforderungs-Analyse/Projektvorschlag	15.04.2021	20.04.2021	5	
1.1.1	Ausgangssituation & Projektbeschreibung	15.04.2021	20.04.2021	1	
1.1.2	Zielgruppen	15.04.2021	20.04.2021	2	
1.1.3	Feature-Liste	15.04.2021	20.04.2021	2	
PVP	Projektvorschlag Präsentation	21.04.2021	21.04.2021	1	
1.2	Projektauftrag erstellen	22.04.2021	03.05.2021	11	
1.2.1	Projektauftrag anpassen	22.04.2021	03.05.2021	3	
1.2.2	Akteurenhierarchie erstellen	22.04.2021	03.05.2021	1	
1.2.3	Icebergliste erstellen	22.04.2021	03.05.2021	4	
1.2.4	groben Projektplan erstellen	22.04.2021	03.05.2021	3	
MS.1	Projektauftrag	03.05.2021	03.05.2021	1	
MR-1	Management Review 1	04.05.2021	04.05.2021	1	
1.3	Anforderungs-Analyse fertigstellen	05.05.2021	08.05.2021	3	
1.3.1	Projektplan verfeinern, Arbeitspakete	05.05.2021	08.05.2021	3	
2.1	Entwurf und Design	05.05.2021	08.05.2021	3	ca, gb
2.1.1	Domänenmodell abschliessen	05.05.2021	08.05.2021	2	ca, sr
2.1.1	Komponentendiagramm fertigstellen	05.05.2021	08.05.2021	1	ca, sr
3	Implementierung Sprint 1	08.05.2021	14.05.2021	6	
3.1	DB Schema & Testdaten	08.05.2021	14.05.2021	6	ak, sr
3.2	Persistenz-Schicht, DAOs, Model Klassen	08.05.2021	14.05.2021	6	ca, sr
3.3	Service Schicht	08.05.2021	14.05.2021	6	gb
3.4	SCM, Tracker Konfiguration	08.05.2021	14.05.2021	6	hk
3.5	Produkt Dokumentation	08.05.2021	14.05.2021	6	ms
3.6	Maven Konfiguration (pom.xml)	08.05.2021	14.05.2021	6	ca, sr

3.7	Tests der Persistenz-Schicht (DAOs)	08.05.2021	14.05.2021	6	ak, sr
MS.2	Erster Systemtest	14.05.2021	14.05.2021	1	
IR-1	Internes Review 1 (Anforderungs Review), Pre-Release Demo	14.05.2021	14.05.2021	1	
2.2	Entwurf und Design fertigstellen	15.05.2021	28.05.2021	13	
3.1	Implementierung Sprint 2 bis Version 1, siehe Iceberglis	18.05.2021	28.05.2021	10	
MS.3	GUI Prototyp mit DAO Tests, 40% der User Stories implementiert	28.05.2021	28.05.2021	1	
MR-2	Management Review 2 (Design Review), Alpha Release Demo	28.05.2021	28.05.2021	1	
3.2	Implementierung Sprint 3 bis Version 2, siehe Iceberglis	29.05.2021	10.05.2021	12	
MS.4	70% der User Stories implementiert, Service-Schicht Tests	10.06.2021	10.06.2021	1	
IR-2	Internes Review (Code Inspektion), Beta Version Demo	11.06.2021	11.06.2021	1	
3.3	Implementierung Sprint 4 bis Version 3, siehe Iceberglis	12.06.2021	23.06.2021	11	
MS.5	100% der User Stories implementiert, Akzeptanztests	24.06.2021	24.06.2021	1	
MR-3	Management Review 3: Projektabnahme, General Availability (GA) Version Demo	25.06.2021	25.06.2021	1	

7.3.2. Meilensteinbeschreibung

Im Folgenden wird kurz angeführt, wie weit die Softwareentwicklung zu den einzelnen Meilensteinen fortgeschritten sein soll.

- Meilenstein 1: Projektauftrag fertig
- Meilenstein 2: Erster Systemtest
Erster Prototyp. Erster Systemtest mit simpler GUI und Datenbankverbindung. CRUD Methoden implementiert und getestet.
- Meilenstein 3: Benutzbare Alpha Version - GUI Prototyp
Zu diesem Zeitpunkt sollen ca. 40% der Userstories implementiert sein - also alle Use Cases, die in der Iceberglis (5.3) Version 1 zugeordnet sind. Weiters sollten bereits Teile der Serviceschicht für Version 2 implementiert und getestet sein.
- Meilenstein 4: Beta Version
Zu diesem Zeitpunkt sollen ca 70% der Userstories implementiert sein; das sind alle, die Version 2 zugeordnet sind. Es sollen außerdem schon die benötigten Serviceschicht-Teile für Version 3 implementiert sein.

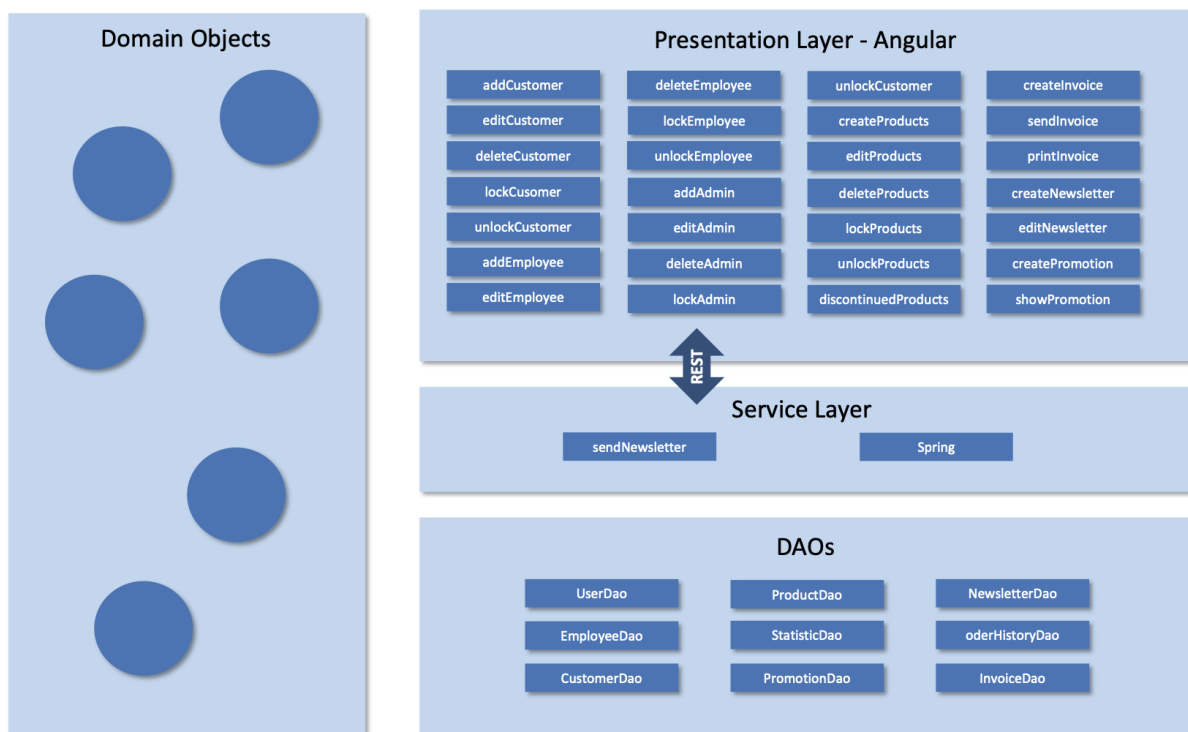
- Meilenstein 5: Final Release
Zu diesem Zeitpunkt sollen alle Use Cases implementiert und getestet (auch auf Akzeptanz) sein. Das Projekt muss funktionieren und vorzeigbar sein.

8. Projektabgrenzung

Folgende Funktionen oder Arbeitspakete liegen nicht im Umfang des Projekts und werden auch nicht vom Entwicklerteam verarbeitet:

- Internationale Transaktionen
- ein vollständiges Buchhaltungssystem, ein Rechnungswertungssystem (Archivierung)
- Verwaltung eingehender Rechnungen und Statistiken bezüglich Gewinn

9. Schichtendiagramm



10. Lieferkomponenten

Während der Projektlaufzeit und vor allem beim Abschluss des Projektes wird dem Kunden die erstellte Software, sowie dazugehörige Dokumentation übergeben. In diesem Abschnitt werden die Softwarekomponenten und Dokumentation beschrieben und eine Abgrenzung des Lieferumfangs durchgeführt.

10.1. Software

Nach Projektabschluss werden dem Kunden folgende Kernkomponenten der Software in produktionsfähigem Zustand übergeben:

- Angular, Angular-Material Frontend
- H2 SQL Datenbank
- Konfigurationsdaten für Datenbanken und Server, soweit für Wartung und Weiterentwicklung notwendig

10.2. Artefakte & Projektdokumentation

- Domänenmodell & Komponentendiagramm
- Funktionale Anforderungen
 - Anwendungsfallbeschreibung
 - Anwendungsfalldiagramme, Pakete & Akteurenhierarchie
- UML Verteilungsdiagramm (Deployment Diagramm)
- Testplan, Funktionale Testfälle, Testberichte (soweit für die Wartung und Weiterentwicklung notwendig)
- Datenbankbeschreibung & ER Diagramm
- Präsentationen zu den Reviews (MRs)

10.3. Meilensteine

Meilenstein 2: Erster Systemtest (Hello World)

- Artefakte des laufenden Projektmanagements (Besprechungsprotokolle, Stundenlisten, Risikoabschätzung, Iceberglst, Projektstrukturplan (Work Breakdown Structure auf Arbeitspaketebene))
- IR-1 Statusbericht
- Funktionale Anforderungen (Übersicht über alle Anwendungsfälle, Beschreibung der Anwendungsfälle, Anwendungsfalldiagramme, Akteur_innenliste, Akteur_innenhierarchie)
- Komponentendiagramm
- UI-Skizze und Beschreibung
- „Hello World“ Prototyp mit GUI: erster Systemtest
- Testartefakte (Testplan)
- Datenbankbeschreibung (ER-Diagramm, Beschreibung der Attribute)

Meilenstein 3: GUI Prototyp mit DAO Tests, 40% der User Stories implementiert

- Artefakte des laufenden Projektmanagements + MR-2 Statusbericht mit Präsentation
- Anwendungsfallbeschreibung überarbeitet und abgeschlossen
- Testplan abgeschlossen
- Datenbankbeschreibung abgeschlossen
- „Alpha Version“: Prototyp mit DAO Tests und 40% der funktionalen Anforderungen (User Stories) implementiert
- UI Skizzen/Design abgeschlossen
- Komponentendiagramm überarbeitet und abgeschlossen
- Testartefakte (Funktionale Testfälle / Akzeptanztests)

Meilenstein 4: 70% der User Stories implementiert, Service-Schicht Tests

- Artefakte des laufenden Projektmanagements + IR-2 Statusbericht
- „Beta Version“: 70% der User Stories implementiert mit Service-Schicht Tests

Meilenstein 5: 100% der User Stories implementiert, Akzeptanztests

- Artefakte des laufenden Projektmanagements + MR-3 Statusbericht mit Präsentation
- Testberichte abgeschlossen
- „General Availability Version“: funktionierendes Produkt

11. Nichtfunktionale Anforderungen

- **Performance** - Jede Operation muss in maximal drei Sekunden zu einem für den User sichtbaren Ergebnis führen.
- **Browser-Konformität** - Die Oberfläche muss zumindest auf Firefox (Version 85) und Chrome/Chromium (Version 88) getestet und ausreichend nutzbar sein um die Features fehlerfrei zu nutzen.
- **Betreiberfreundlichkeit** - Jede Kernfunktionalität (Warenkorb befüllen, Bestellen, Registrierung, Bestellungsverwaltung) soll mit maximal fünf Klicks abgeschlossen werden.
- **Wartbarkeit** - Die Anwendung soll so umgesetzt werden, dass eine spätere Wartung möglichst einfach ausfällt. Dazu gehören die aussagekräftige Kommentierung des Programmcodes sowie ein sauberer Programmierstil.
- **Datensicherheit** - Die Daten der einzelnen Betreiber (Akteure) dürfen nur von den dafür vorgesehenen Administratoren eingesehen werden können.
- **Skalierbarkeit** - Die Software soll auch bei einer hohen Anzahl an Daten effizient funktionieren.

12. Risikoabschätzung

Nr	Typ	P	W	Folgerisiken	Verantwortliche	Name & Beschreibung	Gegenmaßnahmen
1	immer	H	M	2,3	hk, gb	Ausfall eines Projektmitglieds: durch Krankheit oder andere Gründe	Sammeln der bereits geleisteten und noch ausstehenden Arbeit; Besprechung und Aufteilung der Aufgaben innerhalb der Gruppe (zusätzlich zum Jour-Fixe, kann auch elektronisch stattfinden). Zuteilung der Rolle zu einem anderen Gruppenmitglied.
2	entwicklung	H	N		ca, sr	Rechtzeitige Fertigstellung gefährdet: Die rechtzeitige Fertigstellung des	Reduktion von Features anhand ihrer Priorisierung

						Projekts ist gefährdet. Mögliche Ursachen könnten zu hohe Anforderungen, zu geringe zeitliche Ressourcen, Ausfall von Team-Mitgliedern, etc sein.	
3	management	H	N	2	hk, gb	Verlust von projektinternem Wissen: Verlust an projektinternem Wissen durch verschiedene Ursachen (zB Ausscheiden eines Projektmitglieds, Server-Ausfall, etc)	Dokumentation aller relevanten Informationen in Gitlab und regelmäßige Kontrolle
4	planung, entwicklung	H	N	2	ca, sr	Notwendige Libraries nicht verfügbar: Während des Projektverlaufs stellt sich heraus, dass für ein bestimmtes Feature notwendige Libraries nicht (frei) verfügbar sind.	Sofern der Aufwand vertretbar ist, können die Funktionalitäten selbst entwickelt, bzw andere Libraries adaptiert werden; Ist dies nicht möglich, muss das Feature so weit wie notwendig reduziert werden.
5	planung, entwicklung	M	M	2	ak, sr	Unzureichende Design-Entscheidungen: Während der Implementierung stellt sich eine Design-Entscheidung als unzureichend heraus.	Design-Entscheidungen werden so früh wie möglich mit allen Projektmitgliedern besprochen, die die Entscheidung bezüglich Ihrer Rolle beurteilen; Modulares Design, damit die Auswirkungen falscher Design-Entscheidungen möglichst klein gehalten werden

13. Informationswesen

- Wöchentliche Teaminterne online Meetings nach Absprache wenn notwendig
- Interne Kommunikation mittels Discord
- Laufende Kommunikation mit Tutor Adrian Bergler per E-Mail
- 5 Reviews: 2 Interne Reviews mit Tutor_innen, 3 Management Reviews mit Assistent_innen