Projektbericht

Ricardo Furtado de Gois, Fabian Küpper, Pascal Lentz, Martin Fijalkowski, Julius Wartenberg

7.6.2017

Inhaltsverzeichnis

1. Einleitung

Die folgende Projektdokumentation schildert den Ablauf des Mittelstufenprojektes, welches die Autoren im Rahmen ihrer Ausbildung zum Fachinformatiker Fachrichtung Anwendungsentwicklung durchgeführt haben.

1.1 Projektbeschreibung

Der Tante-Emma Laden führt im Moment alle Vorgänge noch manuell mit Papier und Stift durch. Dies macht sich negativ auf die Produktivität der Mitarbeiter aus

In diesem Projekt sollen die gerade beschriebenen Prozesse mithilfe einer Kassen- und Verwaltungssoftware vereinfacht werden.

1.2 Projektziel

Ziel des Projektes ist die Vereinfachung der alltäglichen Vorgänge. Dazu sollen die manuellen Vorgänge durch eine Kassen- und Verwaltungssoftware ersetzt werden. Diese Software wird in Form einer Desktopanwendung mit graphischer Benutzeroberfläche und eigener Datenbank realisiert. Die Anwendung soll die Verwaltung von Kundendaten sowie die Verwaltung von Bestellungen vornehmen. Gleichzeitig sollen auch neue Daten eingepflegt werden können. Beispielsweise sollen neue Artikel in eine Lagerverwaltung eingefügt werden. Außerdem sollen alle Programmbereiche, abhängig von den Berechtigungen des Benutzers, angezeigt werden, oder bestimmte Aktionen nicht zulassen. Alle eingepflegten Daten sollen in der Datenbank automatisch abgelegt werden. Außerdem soll mit der Anwendung auch der alltägliche Kassenbetrieb vereinfacht und beschleunigt werden.

1.3 Auftraggeber

Auftraggeber des Projektes ist die Inhaberin eines Tante-Emma Ladens.

1.4 Projektbegründung

Die Hauptschwachstelle der momentanen Arbeitsprozesse, in dem Tante-Emma Laden, ist das hohe Maß an manueller Arbeit.

Ein Beispiel hierfür wären das Erfassen und Verwalten von Kundendaten oder Bestellungen. Hierbei kann es schnell zu Flüchtigkeitsfehlern kommen, die dann wiederum zu Folgefehlern führen. Außerdem könnte es sogar sein, dass essentielle Daten gar nicht erfasst werden, weil sie einfach vergessen wurden. Des Weiteren muss in regelmäßigen Abständen ein Mitarbeiter alle vorhandenen Bestellungen durchgehen und die Bestellungen der nächsten Tage raussuchen.

Ein weiteres Beispiel hierfür wäre der Kassenbetrieb. Momentan müssen alle Rechnungen noch manuell durchgeführt werden. Dies ist auch sehr fehleranfällig.

Aufgrund dieser Probleme und der dadurch steigenden Fehleranfälligkeit hat sich die Inhaberin des Tante-Emma Ladens dazu entschieden, die Entwicklung einer Anwendung in Auftrag zu geben, durch die alle Arbeitsprozesse vereinfacht und beschleunigt werden sollen.

1.5 Projektabgrenzung

Da der Projektumfang auf 18 Stunden beschränkt ist, soll die Anwendung nicht mit zusätzlicher Hardware kommunizieren können, wie beispielsweise einem Bon Drucker, oder einem Scanner.

2. Projektplanung

2.1 Projektphasen

Für die Umsetzung des Projektes standen den Autoren 18 Stunden (24 Schulstunden) zur Verfügung. Diese wurden vor Projektbeginn auf mehrere Phasen verteilt. Eine Grobe Zeitplanung sowie die einzelnen Phasen lassen sich der Tabelle 1: Grobe Zeitplanung entnehmen.

Projektphase	Geplante Zeit
Projektdefinition	0,5 h
Projektplanung	3,5 h
Projektdurchführung	10 h
Erstellung der Dokumentation	4 h
Gesamt	**18 h**

Tabelle 1: Grobe Zeitplanung

2.2 Ressourcenplanung

In der Übersicht, welche sich im Anhang A.1 Verwendete Ressourcen befindet, sind alle Ressourcen aufgelistet, die für das Projekt benötigt und eingesetzt wurden. Mit den Verwendeten Ressourcen sind sowohl Software- und Hardware- als auch Personalressourcen gemeint. Bei der Auswahl der verwendeten Software wurde darauf geachtet, dass diese kostenfrei zur Verfügung steht, oder bereits Lizenzen für die in A.1 Verwendete Ressourcen genannten Anwendungen schon vorhanden sind. Dadurch sollen anfallenden Projektkosten möglichst minimal ausfallen.

2.3 Entwicklungsprozess

Bevor mit der eigentlichen Realisierung des Projektes begonnen werden konnte, mussten sich die Autoren für einen Vorgehensmodell der Softwareentwicklung entscheiden. Dieses Vorgehensmodell definiert den Ablauf der Umsetzung. Für dieses Projekt wurde von den Autoren das Wasserfall-Modell gewählt. Das Hauptmerkmal dieses Vorgehensmodells ist der lineare Projektablauf. Die nächste Phase im Projekt wird erst begonnen, sobald die vorangehende Phase abgeschlossen ist.

Durch den linearen Ablauf des Projektes und die klare Abgrenzung der einzelnen Phasen ist eine einfache und kontrollierte Planung des Projektes möglich.

In der Projektplanung wurde viel Zeit für die Implementierungsphase einkalkuliert, damit noch ausreichend Zeit für Tests vorhanden war.

2.4 Ist-Analyse

Wie bereits in Abschnitt 1.1 Projektbeschreibung erwähnt müssen momentan alle Arbeitsabläufe noch manuell von dem Personal durchgeführt werden.

Alle Kunden- und Bestellungsdaten müssen momentan noch auf Papier erfasst werden und in einen extra Ordner einsortiert werden. Des Weiteren muss das Personal noch auf einem separatem Dokument notieren, welche Bestellungen in den nächsten Tagen

ausgeliefert werden müssen. In regelmäßigen Abständen muss dieses Dokument auch um neue Bestellungen erweitert werden. Außerdem müssen die Mitarbeiter, um zu schauen welche Artikel noch vorrätig sind, durch den ganzen Laden laufen und die Artikel aus den Regalen raussuchen.

Letztlich werden die Rechnungen auch noch manuell von den Mitarbeitern erstellt.

Diese Arbeitsabläufe sind sehr zeitaufwändig und auch sehr fehleranfällig.

2.5 Soll-Konzept

Es soll eine einfach zu bedienende und übersichtliche Anwendung erstellt werden. Die Anwendung soll eine Startseite haben, in der alltägliche Kunden schnell und einfach abkassiert werden können. Des Weiteren sollen nicht nur Käufe möglich sein, sondern auch Bestellungen erfasst werden. Die Bestellungen sollen auch auf einer eigenen Übersichtsseite dargestellt und eventuell überarbeitet werden können. Das gleiche gilt auch für Kunden. Diese müssen auch in der Anwendung eingepflegt und verwaltet werden können. Jeder Artikel in dem Laden muss auch in das Programm eingepflegt werden können. Außerdem muss auch ersichtlich sein, ob und wie viele Artikel vorrätig sind.

Es sollen auch Benutzer eingerichtet werden, die unterschiedliche Berechtigungen haben. Die Benutzer und deren Berechtigungen, lassen sich auf einer eigenen Übersicht bearbeiten.

Alle unter 2.4 Ist-Analyse beschrieben Vorgänge sollen optimiert und beschleunigt werden.

2.6 Wirtschaftlichkeitsanalyse

Aufgrund der Verzögerungen durch die, in den Abschnitten 1.4 Projektbegründung und 2.4 Ist-Analyse, geschilderten Arbeitsabläufe ist die Umsetzung des Projektes erforderlich. Ob aus wirtschaftlicher Sicht die Realisierung auch gerechtfertigt ist, soll in den folgenden Abschnitten geklärt werden.

2.6.1 Projektkosten

Im Folgenden werden die Projektkosten, die während der Entwicklung des Projektes anfallen, kalkuliert. Dafür werden neben den Personalkosten, die durch die Realisierung des Projektes verursacht werden, auch noch die Ressourcen berücksichtigt. Die Kalkulation wird anhand von Stundensätzen durchgeführt. Der Stundensatz eines Entwicklers beträgt 20 €. Für die Ressourcennutzung wurde ein pauschaler Stundensatz von 10 € festgelegt.

Die jeweiligen Kosten für die einzelnen Vorgänge, sowie die gesamten Projektkosten lassen sich der Tabelle 2: Kostenaufstellung entnehmen.

Vorgang	Mitarbeiter	Zeit	Personal	Ressourcen	Gesamt
Entwicklungskosten	5 x Mitarbeiter	18 h	1.800,00 €	180,00 €	1.980,00 €
Code-Review und Abnahme	5 x Mitarbeiter	1 h	100,00 €	10,00 €	110,00 €
				Projektkosten gesamt	**2.090,00 €**

Tabelle 2: Kostenaufstellung

2.6.2 Amortisationsdauer

In diesem Abschnitt soll ermittelt werden, ab welchem Zeitpunkt sich die Projektkosten wieder amortisiert haben. Anhand des Amortisationswertes kann beurteilt werden, ob die Umsetzung des Projektes aus wirtschaftlicher Sicht sinnvoll ist. Die Amortisationsdauer wird berechnet, indem man die Anschaffungskosten durch die laufende Kostenersparnis teilt.

Durch die Vereinfachung der einzelnen Arbeitsschritte ließe sich Arbeitszeit einsparen. Durch die Einsparung der Arbeitszeit, werden auch gleichzeitig die Personalkosten reduziert. Die Zeitersparnis der einzelnen Vorgänge wird in der Übersicht Tabelle 3: Zeiteinsparung dargestellt. Die angegebenen Zeiten wurden durch die Autoren geschätzt.

Vorgang	Anzahl pro Monat	Zeit (alt) pro Vorgang	Zeit (neu) pro Vorgang	Einsparung pro Monat
Erfassen von Kundendaten	70	4 min	1 min	210 min
Erfassen von Bestellungen	25	4 min	2 min	50 min
Kunden abkassieren	170	2 min	1 min	170 min
			Zeiteinsparung gesamt pro Monat	430 min

Tabelle 3: Zeiteinsparung

Berechnung der Amortisationsdauer:

$$430 \frac{min}{Monat} \text{ x } 12 \frac{Monate}{Jahr} = 5160 \frac{min}{Jahr} = 86 \frac{Std}{Jahr}$$

$$86 \frac{Std}{Jahr} \text{ x } (20€ + 10€) = 2580 \frac{€}{Jahr}$$

$$\frac{2.090{,}00€}{2.580{,}00€} = 0{,}81 \text{ Jahre} = 9 \text{ Monate } 3 \text{ Wochen}$$

Anhand der Amortisationsrechnung ergibt sich für das Projekt eine Amortisationsdauer von 9 Monaten und 3 Wochen. Über diesen Zeitraum muss die Anwendung mindestens eingesetzt werden, damit sich die Anschaffungskosten und die Kosteneinsparung ausgleichen.

Da die Inhaberin des Tante-Emma Ladens diese Anwendung langfristig einsetzen möchte, kann das Projekt auch aus wirtschaftlicher Sicht als sinnvoll erachtet werden.

2.7 Entwurf der Benutzeroberfläche

Um die neue Anwendung möglichst benutzerfreundlich und einfach bedienen zu können, soll eine klar strukturierte Benutzeroberfläche entwickelt werden. Aus der Hauptansicht sollten außerdem auch alle weiteren Programmteile möglichst einfach und schnell erreicht werden können. In der Hauptansicht selber, soll es möglich sein schnell neue Kunden abzukassieren.

Anhand dieser Vorgaben haben wir eine Benutzeroberfläche entwickelt. Der Entwürfe der Benutzeroberflächen befinden sich im Anhang unter A.5 Entwurf der Benutzeroberfläche.

2.8 Datenbank

2.8.1 Ermittlung der Entitäten

Anhand der Informationen die wir beim Kick-Off-Meeting erhalten haben, lassen sich einige Entitäten bilden. Eine zentrale Stelle nehmen der Mitarbeiter und der Kunde ein. Daher haben sich der *Mitarbeiter* und der *Kunde* schon als Entität herausgestellt. Da jeder Mitarbeiter auch Berechtigungen besitzt, gibt es auch die Entität *Rechte.* Eine weitere zentrale Funktion soll die Bestellung von Artikeln sein. Auch hieraus lässt sich wieder schließen, dass es eine Entität *Bestellung* und eine Entität *Artikel* geben muss. Da ein Artikel direkt einer Bestellung zugeordnet werden kann, kann hier auch von einer direkten Beziehung der beiden Entitäten ausgegangen werden. Da eine Bestellung natürlich einen Kunden braucht und einen Mitarbeiter der die Bestellung bearbeitet, lässt sich der Bestellung auch direkt die Beziehung zu den Entitäten *Kunde* und *Mitarbeiter* zuweisen.

Des Weiteren umfasst eine Bestellung nicht nur eine Stückzahl eines Artikels, sondern es können auch mehrere öfter drin vorkommen. Dadurch kann eine weitere Entität ermittelt werden, welches den Artikel, die Bestellung und die jeweilige Menge des Artikels enthält. Diese Entität hat den Namen *Posten*.

Zuletzt wird auch noch eine Entität benötigt, das die vorhandene Menge des Artikels und die Position im Lager enthält. Somit lässt sich der Entität *Artikel* eine weitere Entität zuordnen. Diese Entität haben die Autoren *Bestand* genannt.

2.8.2 Datenmodell

Auf Grundlage der herausgearbeiteten Entitäten in 2.8.1 Ermittlung der Entitäten wurde ein ERD erstellt, welches die Beziehungen der einzelnen Entitäten noch einmal graphisch veranschaulichen soll. Das ERD befindet sich im Anhang A.4 Entity-Relationship-Model.

2.9 Geschäftslogik

Die Klassen der Anwendung wurden aufgrund der in 2.5 Soll-Konzept festgelegten Anforderungen und der in 2.8.1 Ermittlung der Entitäten ermittelten Werte bestimmt. Das vollständige Klassendiagramm der Anwendung befindet sich im Anhang unter A.6 Klassendiagramm. Das Klassendiagramm veranschaulicht die einzelnen Klassen der Anwendung und ihre Beziehung zueinander. Dieses Klassendiagramm dient als Grundlage für die Implementierung der Geschäftslogik.

2.10 Pflichtenheft

Am Ende der Planungsphase wurde mithilfe des Lastenhefts, und der Vorarbeit aus der Planungsphase, das Pflichtenheft verfasst. Das Pflichtenheft diente als Leitfaden für die Projektdurchführung.

Das Pflichtenheft befindet sich im Anhang A.2 Pflichtenheft.

3 Projektdurchführung

3.1 Wahl der Art des Programms

Da für die Nutzer eine Anwendung geschaffen werden soll, welche die Informationen auf einfache Weise darstellen soll, fiel die Wahl auf eine Windows Forms Anwendung. Als Alternative für die Windows Forms Anwendung gab es noch eine Webanwendung sowie eine Konsolenanwendung.

Der Vorteil einer Windows Forms Anwendung ist, dass die Entwicklungszeit für eine graphische Benutzeroberfläche wesentlich kürzer ist und die Erstellung auch einfacher ist, als bei einer Webanwendung. Des Weiteren ist eine Windows Forms Anwendung auch schneller als eine Webanwendung.

Außerdem lässt sich eine Windows Forms Anwendung, bei entsprechender Gestaltung, intuitiver bedienen als eine Konsolenanwendung.

3.2 Entwicklungsumgebung

Das Projekt wurde auf einem Rechner mit Windows 7 als Betriebssystem entwickelt. Neben dem aktiviertem Betriebssystem, wird außerdem noch eine MySQL-Datenbank genutzt.

Die eigentlich Anwendung wurde mit Microsoft Visual Studio 2013 in C# entwickelt.

3.3 Implementierung der Datenstrukturen

Auf Basis der Strukturen, die bereits in Abschnitt 2.8 Datenbank für die Anwendung definiert wurden, wurde die Datenbank implementiert. Als erstes wurde eine neue Datenbank erstellt. Diese Datenbank wurde von den Autoren *Emma* genannt. Anschließend wurden die benötigten Tabellen manuell über ein SQL-Skript erstellt. Dieses Skript wurde im Vorfeld von den Autoren vorbereitet. Das verwendete SQL-Skript befindet sich im Anhang unter A.7 SQL-Skript zur Erstellung der Datenbank.

Die Datenbank entspricht somit der in der Projektplanung definierten Struktur und erfüllt die nötigen Anforderungen.

3.4 Implementierung der Geschäftslogik

Da die Implementierung der Geschäftslogik der Hauptbestandteil des ganzen Projektes ist, soll diese in dem Abschnitt genauer erläutert werden. Um die Implementierung der Geschäftslogik möglichst komfortabel zu gestalten ist die Entwicklungsumgebung ein wichtiger Faktor. Dies lässt sich dem Abschnitt 3.2 Entwicklungsumgebung entnehmen.

Zu Beginn der Projektdurchführung wurde zunächst das Grundgerüst des Projektes erstellt. Dazu wurde zunächst ein neues Projekt in der Entwicklungsumgebung erstellt. Die Struktur innerhalb des Projektes wurde anhand der GUI-Elemente in dem Projekt sortiert. Beispielsweise wurden alle für die Bestellung benötigten Klassen zu der Bestellübersicht sortiert, oder alle relevanten Klassen für Benutzer und den entsprechenden Rechten wurden zur Benutzerverwaltung sortiert. Ein Beispiel hierfür ist im Anhang unter A.8 Projektmappenverzeichnis.

Analog zu den einzelnen Klassen der Geschäftslogik wurde auch noch ein Interface definiert. Das Interface *IDBMethoden* definiert die Kommunikation zur Datenbank und erleichtert somit die Entwicklung der für die Geschäftslogik relevanten Klassen. Alle Datenbank-Tabellen sind gleichzeitig auch Klassen in der Anwendung, die dieses Interface implementiert haben.

3.5 Implementierung der Benutzeroberfläche

Die Implementierung der Benutzeroberfläche wurde mit Hilfe des .NET Frameworks durchgeführt. Die Benutzeroberflächen wurden anhand der Entwürfe im Anhang A.5 Entwurf der Benutzeroberfläche gestaltet. Die Funktionalitäten die sich hinter den einzelnen GUI-Elementen befinden, wurden mit Hilfe von Events realisiert. Dies bedeutet, dass nur bei bestimmten Anwender-Aktionen der Code ausgeführt wird. Da die Methoden der einzelnen Klassen bereits vor der Implementierung der Benutzeroberfläche getestet wurden, mussten nur noch die Abläufe innerhalb der GUI-Elemente selber getestet werden.

3.6 Probleme während der Implementierung

Während der Implementierung der Benutzeroberfläche ist den Autoren im Rahmen der regelmäßigen Tests aufgefallen, dass die Zeit nicht mehr ausreichend war um einige Funktionen noch zu verbessern.

Die Funktion, Bestellungen zu stornieren/löschen, musste sogar aufgrund mangelnder Zeit komplett rausgenommen werden. Allerdings wurden schon einige Ansätze bezüglich dieser Funktion im Programmcode realisiert, sodass diese Funktion jederzeit relativ zeitnah realisiert werden kann.

4 Qualitätsmanagement

4.1 Manuelle Test

Vor der eigentlichen Erstellung der Anwendung wurde eine Testliste geschrieben, mit der die einzelnen vorgegebenen Funktionen auf ihre Richtigkeit überprüft werden konnten.

Während der Erstellung der Anwendung wurde darauf geachtet, dass die Funktionen, die in der Testliste aufgeführt sind, auch funktionieren. Dies wurde durch die regelmäßige Überprüfung der einzelnen Funktionen gewährleistet.

Die Testliste mit den einzelnen Testpunkten befindet sich im Anhang unter A.3 Manuelle Testliste

Bei den abschließenden Tests ist schließlich aufgefallen, dass die Funktion „Bestellungen stornieren“ nicht komplett vorhanden war. Dieser Punkt wurde auch noch einmal in dem Abschnitt 3.6 Probleme während der Implementierung beschrieben.

Bei den restlichen Testpunkten wurden keine weiteren Mängel mehr festgestellt.

5 Projektabschluss

5.1 Soll- Ist-Vergleich

Bei einer rückblickenden Betrachtung des Projektes, kann festgehalten werden, dass alle zuvor festgelegten Anforderungen gemäß dem Pflichtenheft (A.2 Pflichtenheft) erfüllt wurden. Die Zeitplanung, die in dem Abschnitt 2.1 Projektphasen festgelegt wurde, konnte auch eingehalten werden. In der Tabelle 4: Soll-/Ist-Vergleich wird die tatsächliche Zeit, die für das Projekt benötigt wurde, der zuvor eingeplanten Zeit gegenübergestellt. Es ist zu erkennen, dass es keine Abweichungen von der Zeitplanung gab.

Projektphase	Soll	Ist	Differenz
Analysephase	0,5	0,5	0
Entwurfsphase	3,5	3,5	0
Implementierungsphase	10	10	0
Erstellung der Dokumentation	4	4	0
Gesamt	**18 h**		**0**

Tabelle 4: Soll-/Ist-Vergleich

Glossar

MySQL	Relationales Datenbankmanagementsystem
GUI	Graphical User Interface
ERD	Entity-Relationship-Diagram

A.1 Verwendete Ressourcen

Hardware

- Arbeitsplätze mit Rechnern
- Tastatur
- Maus

Software

- Windows 7 – Betriebssystem
- Visual Studio 2013 – Entwicklungsumgebung C#
- MySQL – Datenbanksystem
- MySQL Workbench – Verwaltungswerkzeug für MySQL-Datenbanken
- UMLET – Tool zur Erstellung von Diagrammen (Klassendiagramm, Sequenzdiagramm, usw.)

Personal

- 5 Entwickler – Umsetzung des Projektes

A.2 Pflichtenheft

Gruppe:

Gruppenmitglieder:

Fabian Küpper (Prozessbeobachter),

Ricardo Furtado de Gois (Projektleiter),

Pascal Lentz (Qualitätsmanager),

Julius Wartenberg (Chef-Entwickler),

Martin Fijalkowski (Chef-Entwickler)

Auftrag:

Es muss ein Kassenprogramm für den Einsatz in einem Tante Emma Laden konzipiert und erstellt werden. Dieses Kassenprogramm soll unteranderem auch eine Benutzerverwaltung und einer eingeschränkte Lagerverwaltung besitzen. Mit dem Programm sollen auch Bestellungen von Kunden verwaltet werden können.

Inhaltsverzeichnis

1 Zielbestimmung

1.1 Musskriterien

1.1.10 - Das Programm soll alle im Laden verfügbaren Artikel verwalten können.

1.1.20 – Es soll möglich sein, Artikel ins Lager einzupflegen (Zugelieferte Artikel)

1.1.30 – Gleiche Artikel können mehrfach im Lager vorhanden sein.

1.1.40 – Das Programm soll auch normale Ladenverkäufe abwickeln können

(Anonymer Kunde)

1.1.50 – In dem Programm sollen Mitarbeiter erfasst und verwaltet werden können.

1.1.60 – In dem Programm sollen Lieferkunden erfasst und verwaltet werden können.

1.1.70 - Benutzerverwaltung mit Rechten und Rollenverteilung.

1.2 Wunschkriterien

1.2.10 – Login-System für jeden Benutzer

1. 3 Abgrenzungskriterien

1.3.10 – Die Daten müssen nicht verschlüsselt werden.

2 Produkteinsatz

2.1 Anwendungsbereiche

Das Programm ist für den alltäglichen Gebrauch im Kassenbereich eines kleinen Tante Emma Ladens.

2.2 Zielgruppen

Diese Anwendung ist für den Einsatz des Kassenpersonals des Tante Emma Ladens einsetzbar, welche die Einführung in das Programm durch eine Einweisung erhalten haben.

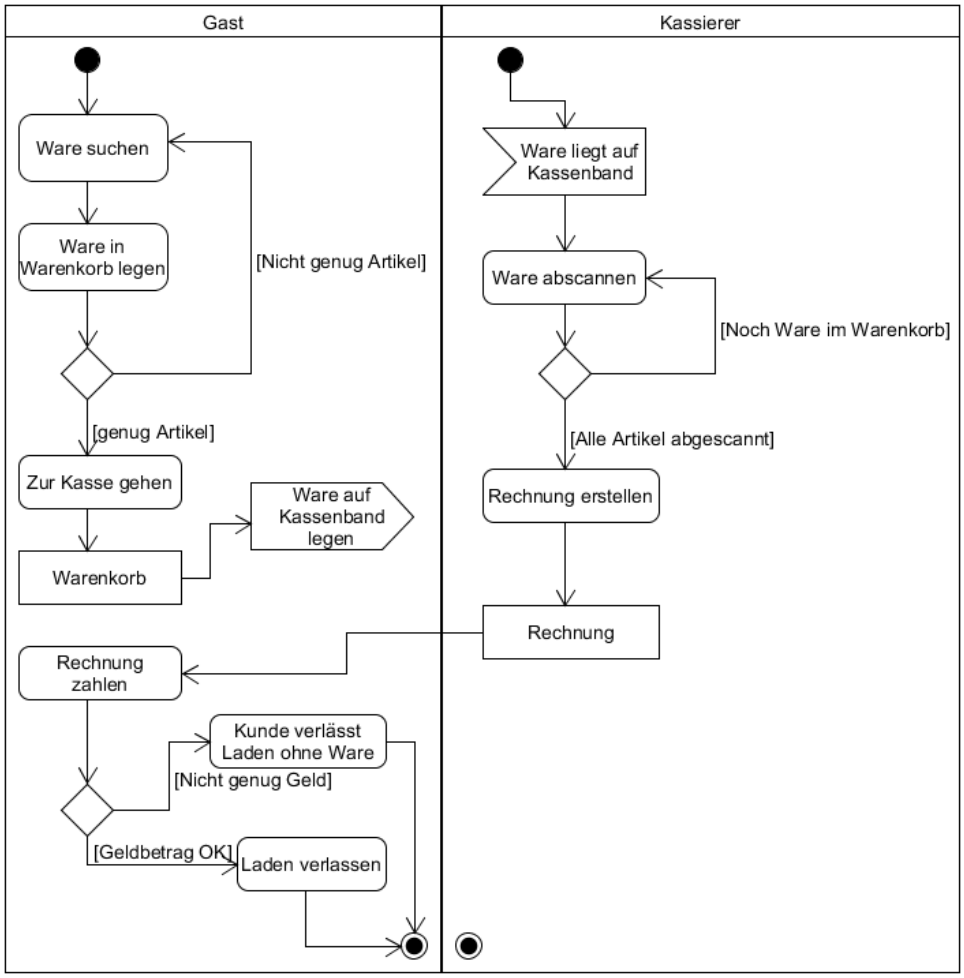
Es werden Basiskenntnisse in EDV Grundlagen vorausgesetzt.

Zusätzlich werden Grundkenntnisse in Deutsch benötigt, um die Anwendung verwenden zu können.

2.3 Betriebsbedingungen

- Betriebsdauer: Werktage (Mo-Sa), ~9 Stunden
- Die Sicherung der einzelnen Daten innerhalb der Datenbank muss vom Administrator durchgeführt werden.
- Der Anwender sollte eine Schulung zur Anwendung dieses Programms erhalten haben.
- Die Hauptzielgruppe sind die Kassierer-/innen des Ladens.
- Der Einsatz der Anwendung erfolgt im Verkaufsraum des jeweiligen Ladens.

3 Produktübersicht



(Aktivitätsdiagramm: Gesamter Geschäftsprozess Anonymer Kunde)

4 Produktfunktionen

/F00/ IDBMethoden

- /F01/ DeleteDB
- /F02/ InsertDB
- /F03/ LoadData
- /F04/ UpdateDB

/F10/ Benutzer

- /F12/ GetAllBenutzer

/F30/ WarenkorbEintrag

- /F31/ GetByVorgangId

/F40/ Bestellung

- /F41/ GetBestellungen

/F50/ Artikel

- /F51/ GetAllArtikel

/F60/ Bestand

- /F61/ GetListByArtikelId

/F70/ Kunde

- /F71/ GetAllKunden
- /F72/ Kunde

/F80/ Mitarbeiter

/F90/ Rechte

5 Produktdaten

Langfristig gespeichert werden Kundendaten, Lagerdaten und Benutzerdaten.

Kundendaten beinhalten:

Vorname, Name, Geburtsdatum, Straße, Hausnummer, Postleitzahl, Ort.

Lagerdaten beinhalten:

Name, Preis, Artikeltyp, Artikelnummer, Menge

Benutzerdaten:

Benutzername, Passwort, Vorname, Name, Geburtsdatum, Straße , Hausnummer,

Postleitzahl, Ort

6 Qualitätsanforderungen

Produktqualität	sehr gut	gut	normal	nicht relevant
Funktionalität				
Sicherheit				x
Ordnungsmäßigkeit			x	
Richtigkeit		x		
Zuverlässigkeit				
Fehlertoleranz			x	
Wiederherstellbarkeit			x	
Benutzbarkeit				
Verständlichkeit	x			
Erlernbarkeit	x			
Bedienbarkeit	x			

7 Benutzungsoberfläche

Das Programm sollte eine möglichst einfache Oberfläche haben. Es sollte intuitiv bedienbar sein und für den Anwender nicht komplex wirken.

Hauptsächlich wird das Programm mit der Maus bedient. Vereinzelt sind auch Tastatureingaben gewünscht.

Hilfskraft – Leseberechtigungen

Personal – Leseberechtigungen, eingeschränkte Schreibberechtigungen

Chef – Vollzugriff in Lese- u. Schreibberechtigungen

8 Nichtfunktionale Anforderungen

Als Sicherheitsanforderung besitzt das Programm sowie die MySQL Datenbank einen Passwortschutz.

Durch das Mitlaufen des Änderungsprotokolls, werden alle Ereignisse des Programms innerhalb der MySQL Datenbank mitgeloggt und ist nur für die Chef-Rolle einsehbar.

Die Anwendung setzt folgende Unterstützte Betriebssysteme voraus:

- Windows 7, 32/64 Bit
- Windows 8, 32/64 Bit
- Windows 8.1, 32/64 Bit
- Windows 10, 32/64 Bit

9 Technische Produktumgebung

9.1 Software

Auf der Zielmaschine wird Empfohlen eine Windows Version ab Windows 7 aufwärts zu besitzen, außerdem muss das Betriebssystem das .NET Framework 4.0 unterstützen und Installiert haben.

Zusätzlich muss eine MySQL Datenbank Installiert und Eingerichtet werden.

- Client
    - .**NET Framework** ab Version 4.0
    - **Windows** fähiger Rechner
- Server
    - **MySQL**-Datenbank

9.2 Hardware

- Client
    - Internetfähiger Rechner
- Server
    - Internetfähiger Server
    - Der Server sollte die Anforderungen einer MySQL Datenbank erfüllen.
    - Ausreichend Rechen- und Festplattenkapazität.

10 Spezielle Anforderungen an die Entwicklungs-Umgebung

10.1 Software

Aufgrund der vielen Komfortfunktionen haben wir uns für Visual Studio 2013 Professional entschieden.

Zusätzliche sollte die Entwicklungs-Umgebung die Hochsprache C# unterstützen und einen Compiler beinhalten.

Außerdem sollte Git-Hub innerhalb von Visual Studio aus verwendbar sein.

10.2 Hardware

Hardwareanforderungen

- 1,6-GHz-Prozessor oder schneller
- 1 GB RAM (1,5 GB bei Ausführung auf einem virtuellen Computer)
- 20 GB verfügbarer Festplattenspeicher
- Festplattenlaufwerk mit 5400 U/min
- DirectX 9-kompatible Grafikkarte mit einer Auflösung von 1024x768 oder höher

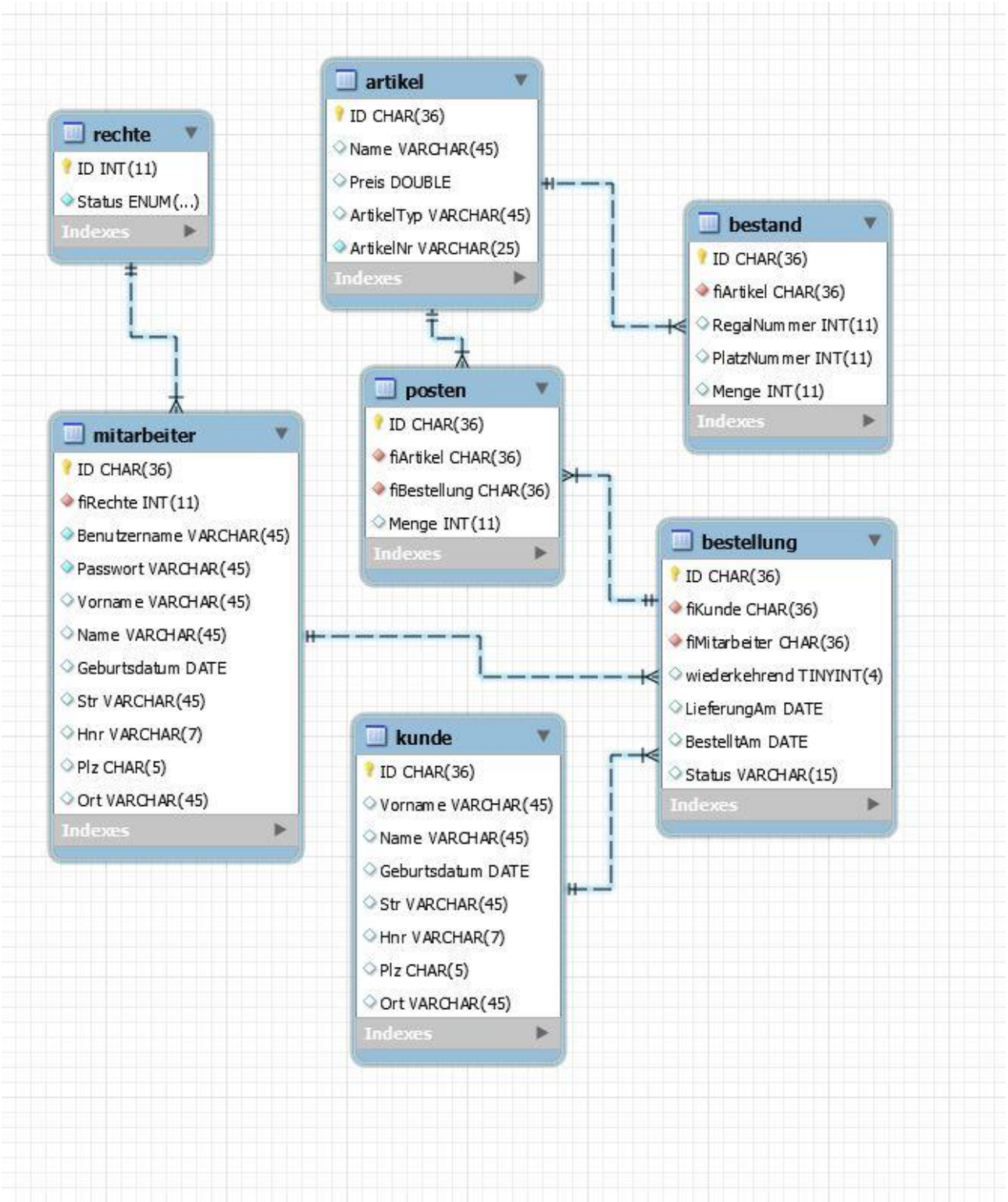
11 Ergänzungen

Eine Bereitstellung der Testdaten, erfolgt über eine beigelegte SQL-Datei und ist bei Bedarf über MYSQL Workbench einzuspielen.

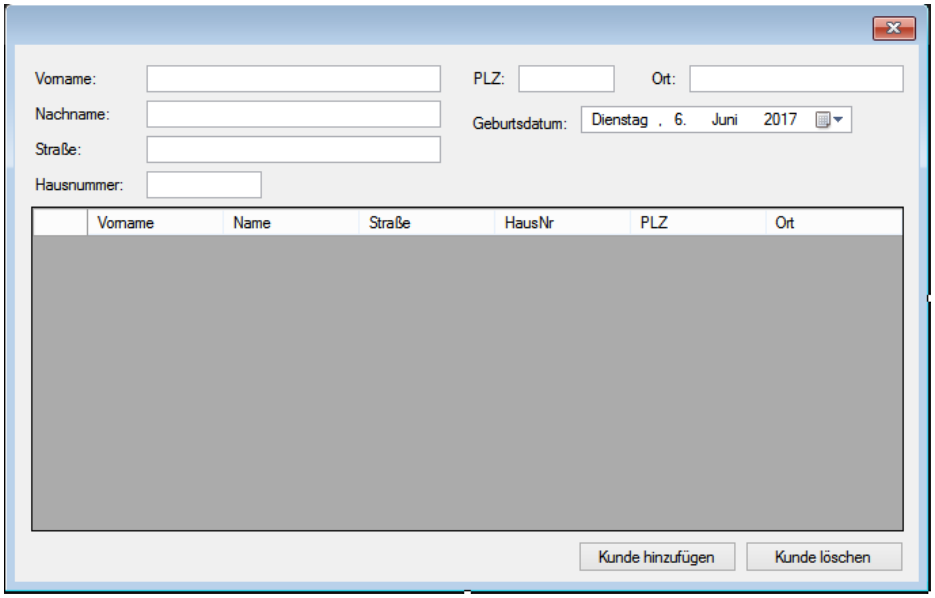
A.3 Manuelle Testliste

Testschritt	Soll-Spezifikation	Ergebnis	Status
1	Das Programm sollte in der Lage sein, einen Kunden zu löschen, neu anzulegen, oder auch zu bearbeiten.	✓	Getestet
2	Es muss eine Übersicht geben, in der Alle Kunden aufgelistet sind.	✓	Getestet
3	Es müssen Bestellungen erfasst werden können.	✓	Getestet
4	Die Bestellungen müssen in einer Übersichtsliste aufgelistet werden.	✓	Getestet
5	Die Bestellungen müssen auch bearbeitet, oder gelöscht/storniert werden können.	X	Getestet
6	Eine Bestellung muss einem Kunden zugeordnet werden können.	✓	Getestet
7	Es muss eine Übersicht geben, in der alle Artikel und alle vorrätigen Artikel aufgelistet werden.	✓	Getestet
8	Es muss eine Kassen-Übersicht geben, in der normale Kunden abkassiert werden können.	✓	Getestet
9	Es muss eine Übersicht geben, in der alle Benutzer/Mitarbeiter aufgelistet sind.	✓	Getestet
10	Jeder Mitarbeiter/Benutzer muss Rechte besitzen.	✓	Getestet
11	Die einzelnen Funktionen der Anwendung müssen von der Berechtigung des Benutzers/Mitarbeiters abhängig sein.	✓	Getestet
12	Beim Verkauf, oder bestellen, von Artikeln muss automatisch ein Lagerabgang verbucht werden.	✓	Getestet
13	Die Anwendung muss eine Lager-Übersicht haben, in der ein Lagerzugang verbucht werden kann.	✓	Getestet
14	Es sollen nur vorhandene Artikel gekauft oder bestellt werden können.	✓	Getestet

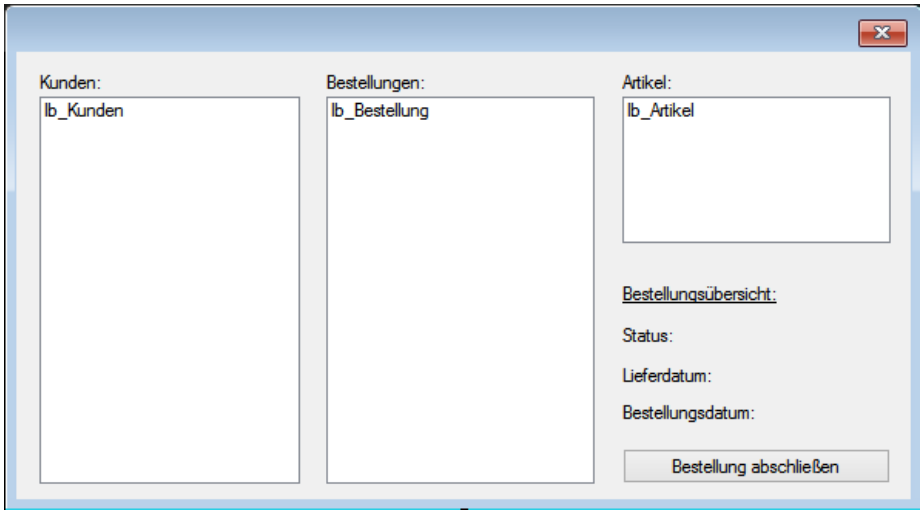
A.4 Entity-Relationship-Model



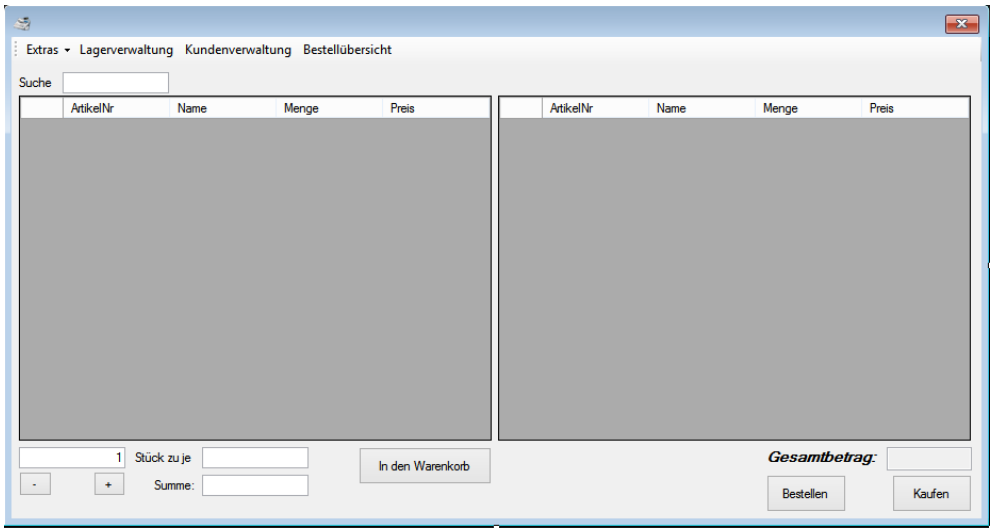
A.5 Entwurf der Benutzeroberfläche



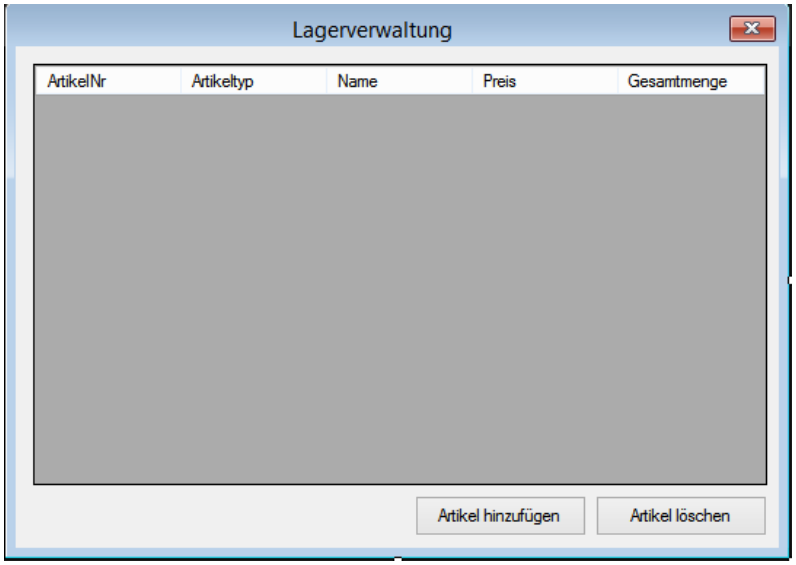
Entwurf der Kundenverwaltung



Entwurf der Bestellungsübersicht

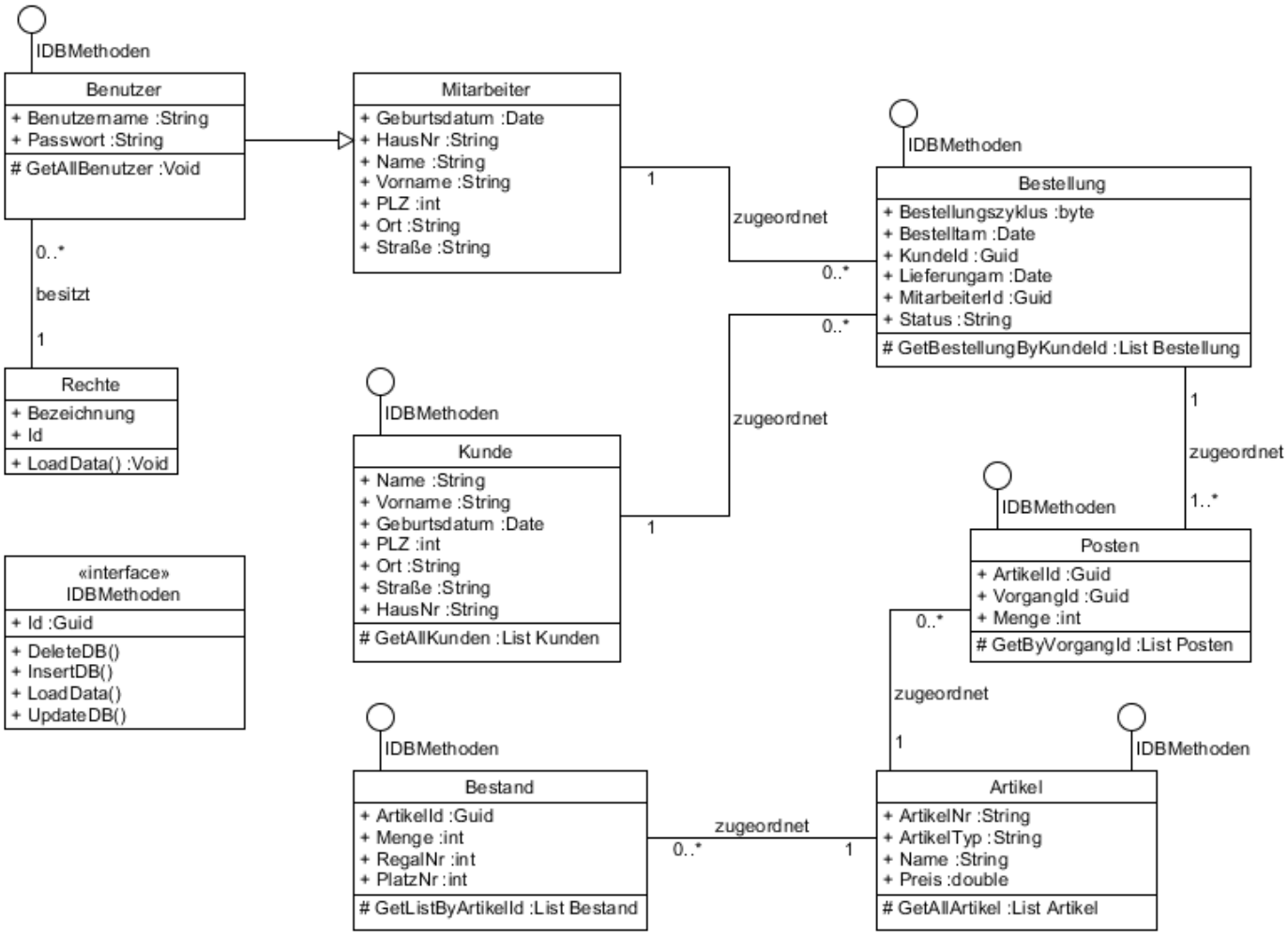


Entwurf der Hauptseite



Entwurf der Lagerverwaltung

A.6 Klassendiagramm



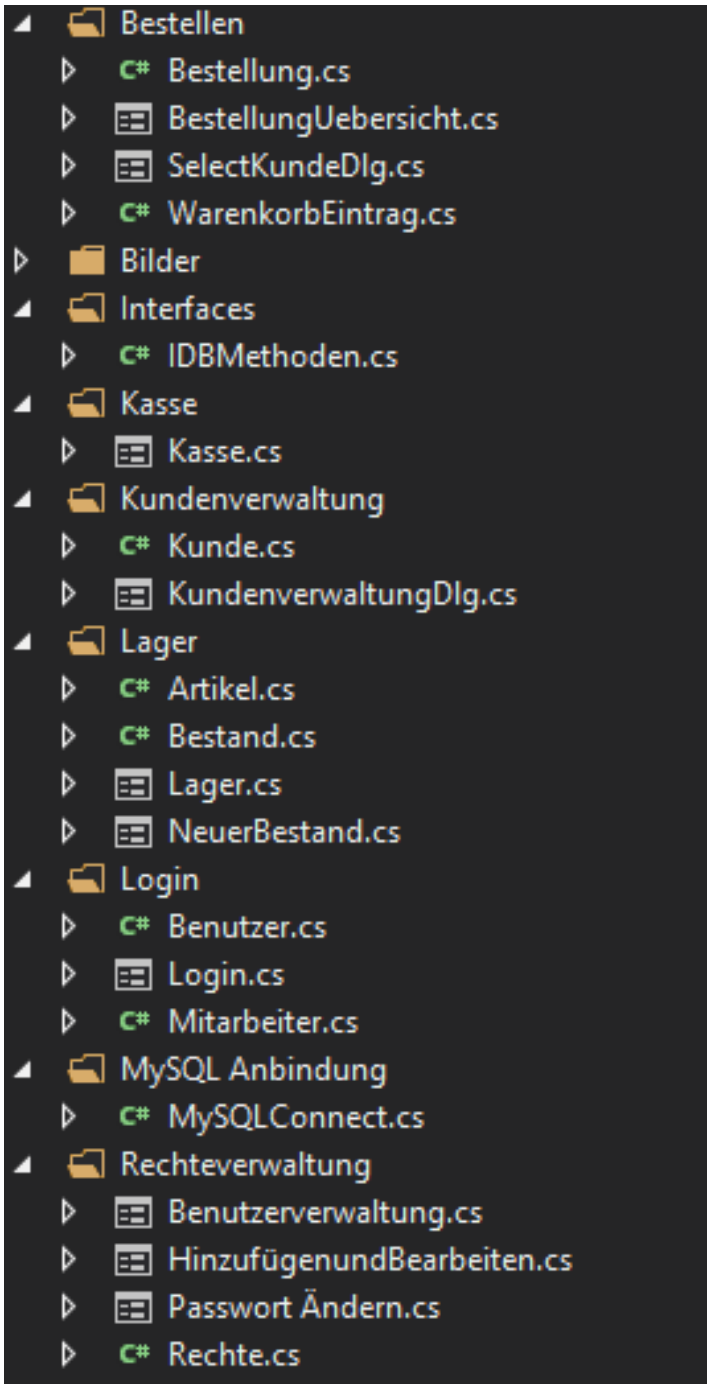
A.7 SQL-Skript zur Erstellung der Datenbank

```
DROP DATABASE IF EXISTS EMMA;

CREATE DATABASE EMMA;

USE EMMA;


DROP TABLE IF EXISTS Rechte;

DROP TABLE IF EXISTS Artikel;

DROP TABLE IF EXISTS Bestand;

DROP TABLE IF EXISTS Bestellung;

DROP TABLE IF EXISTS Posten;

DROP TABLE IF EXISTS Person;

DROP TABLE IF EXISTS Mitarbeiter;


CREATE TABLE `Rechte` (
  `ID` int(11) NOT NULL,
  `Status` enum('Hilfskraft','Personal','Chef') NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB;


CREATE TABLE `Kunde` (
  `ID` char(36) NOT NULL,
  `Vorname` varchar(45) DEFAULT NULL,
  `Name` varchar(45) DEFAULT NULL,
  `Geburtsdatum` DATE ,
  `Str` varchar(45) DEFAULT NULL,
  `Hnr` varchar(7) DEFAULT NULL,
  `Plz` char(5) DEFAULT NULL,
  `Ort` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB ;
```

```
CREATE TABLE `Mitarbeiter` (
  `ID` char(36) NOT NULL,
  `fiRechte` int(11) NOT NULL,
  `Benutzername` varchar(45) NOT NULL,
  `Passwort` varchar(45) NOT NULL,
  `Vorname` varchar(45) DEFAULT NULL,
  `Name` varchar(45) DEFAULT NULL,
  `Geburtsdatum` DATE ,
  `Str` varchar(45) DEFAULT NULL,
  `Hnr` varchar(7) DEFAULT NULL,
  `Plz` char(5) DEFAULT NULL,
  `Ort` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  KEY `fiRechte` (`fiRechte`),
 FOREIGN KEY (`fiRechte`) REFERENCES `rechte` (`ID`) ON DELETE CASCADE
) ENGINE=InnoDB;


CREATE TABLE `Artikel` (
  `ID` char(36) NOT NULL,
  `Name` varchar(45) NULL,
  `Preis` double DEFAULT NULL,
  `ArtikelTyp` varchar(45) DEFAULT NULL,
  `ArtikelNr` varchar(25) NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE=InnoDB;

CREATE TABLE `Bestand` (
  `ID` char(36) NOT NULL,
  `fiArtikel` char(36) NOT NULL,
  `RegalNummer` int(11) DEFAULT NULL,
  `PlatzNummer` int(11) DEFAULT NULL,
```

```
  `Menge` int(11) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  CONSTRAINT `bestand_ibfk_1` FOREIGN KEY (`fiArtikel`) REFERENCES `artikel` (`ID`) ON DELETE CASCADE
) ENGINE=InnoDB ;


CREATE TABLE `Bestellung` (
  `ID` char(36) NOT NULL,
  `fiKunde` char(36) NOT NULL,
  `fiMitarbeiter` char(36) NOT NULL,
  `wiederkehrend` tinyint DEFAULT NULL,
  `LieferungAm` date DEFAULT NULL,
  `BestelltAm` date DEFAULT NULL,
  `Status` varchar(15) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  KEY `fiKunde` (`fiKunde`),
  KEY `fiMitarbeiter` (`fiMitarbeiter`),
  CONSTRAINT `bestellung_ibfk_1` FOREIGN KEY (`fiKunde`) REFERENCES `kunde` (`ID`) ON DELETE CASCADE,
  CONSTRAINT `bestellung_ibfk_2` FOREIGN KEY (`fiMitarbeiter`) REFERENCES `mitarbeiter` (`ID`) ON DELETE CASCADE
) ENGINE=InnoDB;

CREATE TABLE `Posten` (
  `ID` char(36) NOT NULL,
  `fiArtikel` char(36) NOT NULL,
  `fiBestellung` char(36) NOT NULL,
  `Menge` int(11) DEFAULT NULL,
  PRIMARY KEY (`ID`),
  KEY `fiArtikel` (`fiArtikel`),
  KEY `fiBestellung` (`fiBestellung`),
  CONSTRAINT `posten_ibfk_1` FOREIGN KEY (`fiArtikel`) REFERENCES `artikel` (`ID`) ON DELETE CASCADE,
  CONSTRAINT `posten_ibfk_2` FOREIGN KEY (`fiBestellung`) REFERENCES `bestellung` (`ID`) ON DELETE CASCADE
) ENGINE=InnoDB;
```

A.8 Projektmappenverzeichnis



A.9 Fachkonzept

Fachkonzept

Ergebnis der Anforderungsanalyse.

Inhaltsangabe

Vorwort

Diese Anlage zum Pflichtenheft gibt dem Auftragsgeber Aufschluss über unser Verständnis der Anforderung. Dadurch werden Missverständnisse und vermeintliche Fehlimplementationen, die nicht dem Raster dieser Konzeption folgen, sowie Änderungswünsche seitens der Auftragsgeber während der Umsetzung und nach Projektabschluss ausgeschlossen.

Die folgenden Dokumentationen sind fachlich zu verstehen. Es werden fachlich die Geschäftsprozesse des Auftragsgebers beschrieben und festgehalten.

Die Diagramme beschreiben alle Kernfunktionen aus dem Pflichtenheft grob. Dabei verzichten wir bewusst, weitere besondere Umstände einer Kundenabwicklung einzubeziehen. Besondere Umstände sind praktisch selbstverständlich und auf Grund der Vielfalt an möglichen Szenarien nicht zu beschreiben. Die technische realisierung erfolgt weitestgehend ohne Einbeziehen des Auftraggebers.

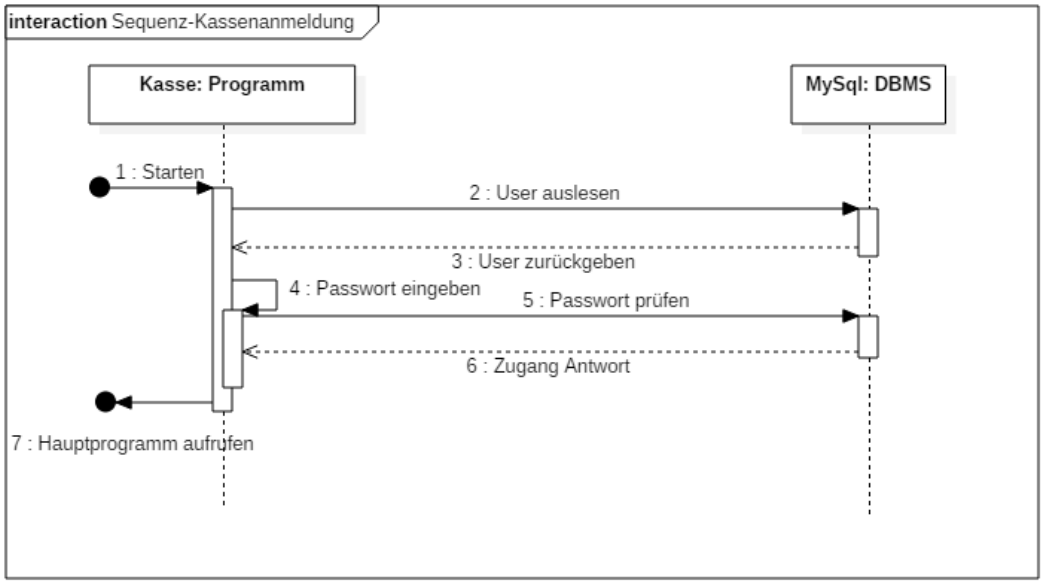
I. Authentifizierung

Kassenanmeldung

Die Kasse wird sicherheitsbedingt nur durch Authentifiziertes Kundenpersonal bedient. Das Zusammenspiel mit der Datenbank, sowie die Eingabe des Kennwortes wird im

folgenden Diagramm dargestellt. Eine Übereinstimmung mit der Benutzereingabe führt zu einer gelungenen Authentifizierung.

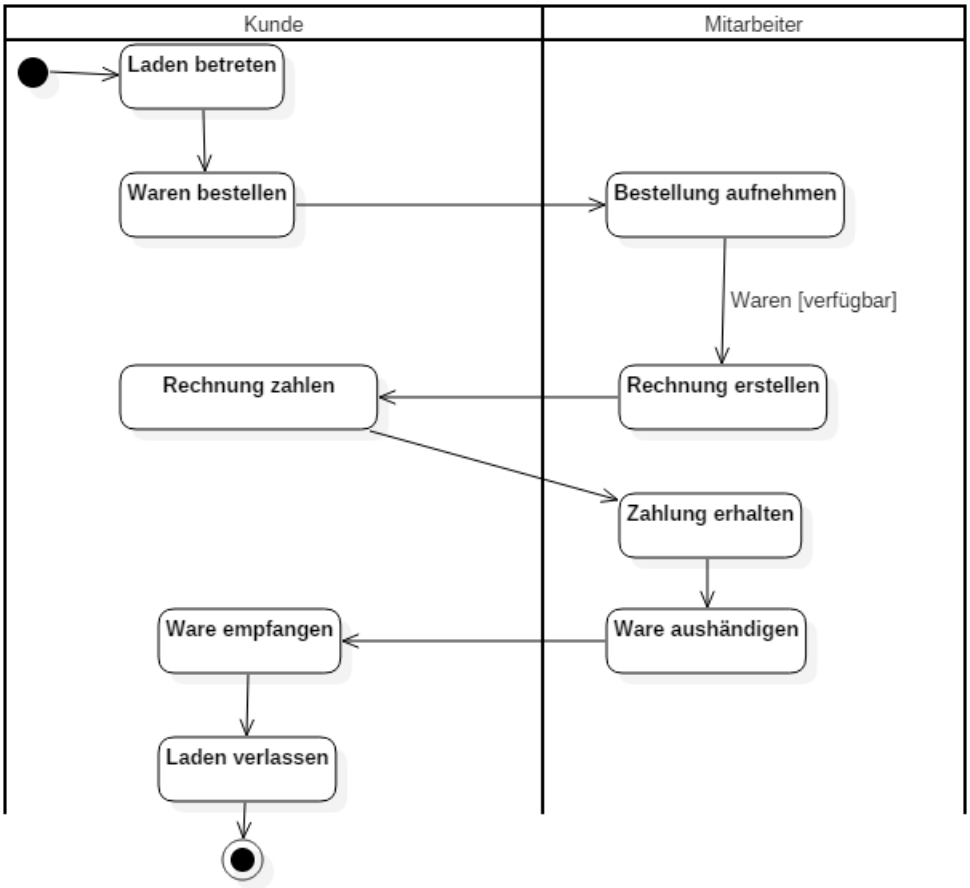
Die Authentifizierung eröffnet die Möglichkeit Funktionen, unter Berücksichtigung der Benutzergruppe und der damit eingeräumten Zugriffsberechtigungen, zu verwenden.



II. Geschäftsprozess

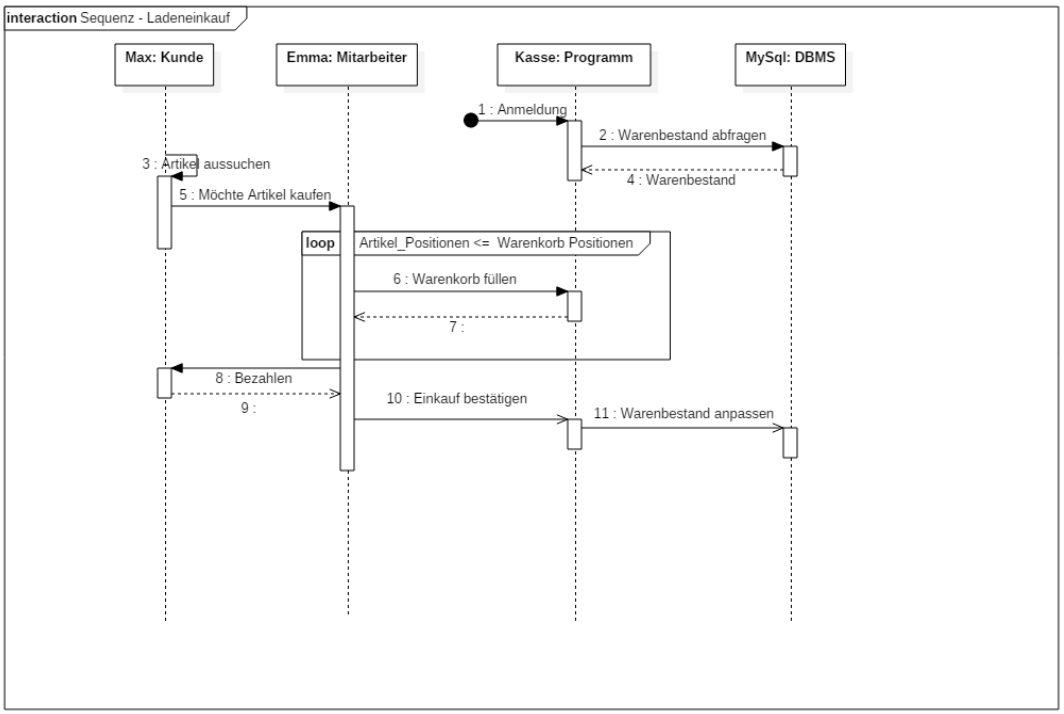
Kundenabwicklung

Die folgende Grafik beschreibt den Ablauf von einer Bestellung bis hin zur Auslieferung der Ware. Die Kasse wird immer von Kassenpersonal bedient.



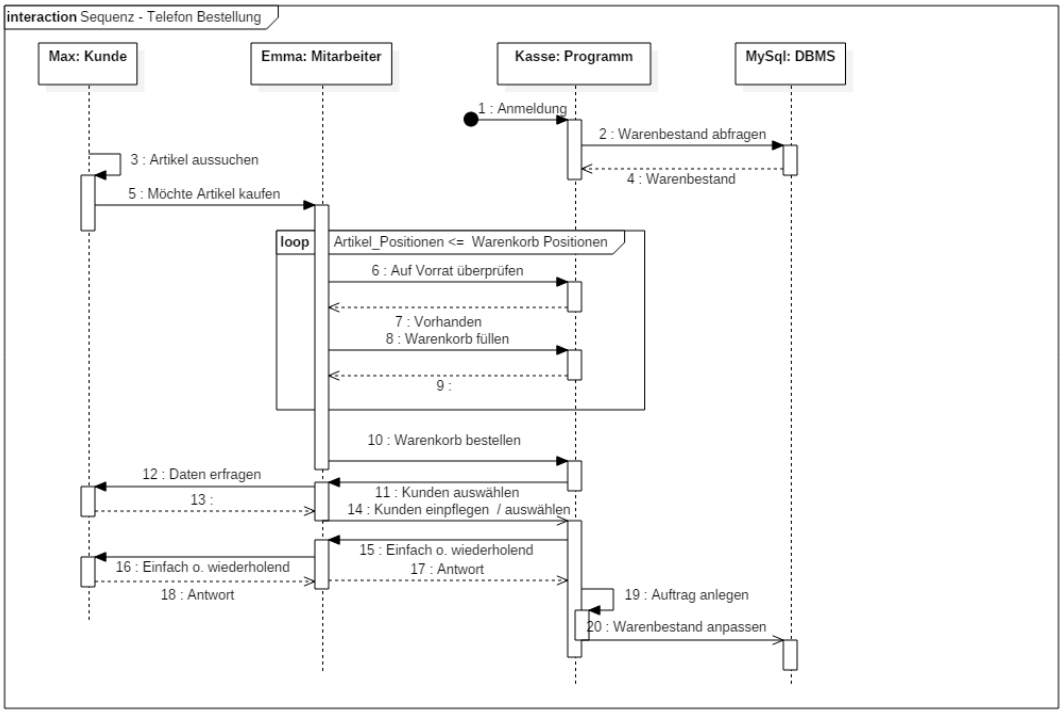
Ladeneinkauf

Das Folgende Diagramm zeigt Tante Emma, als authentifiziertes Kassenpersonal. Die Waren werden mündlich an das Personal übermittelt. Aus dem daraus entstandenen Warenkorb wird ein üblicher Zahlungsvorgang abgewickelt.



Telefon Bestellung

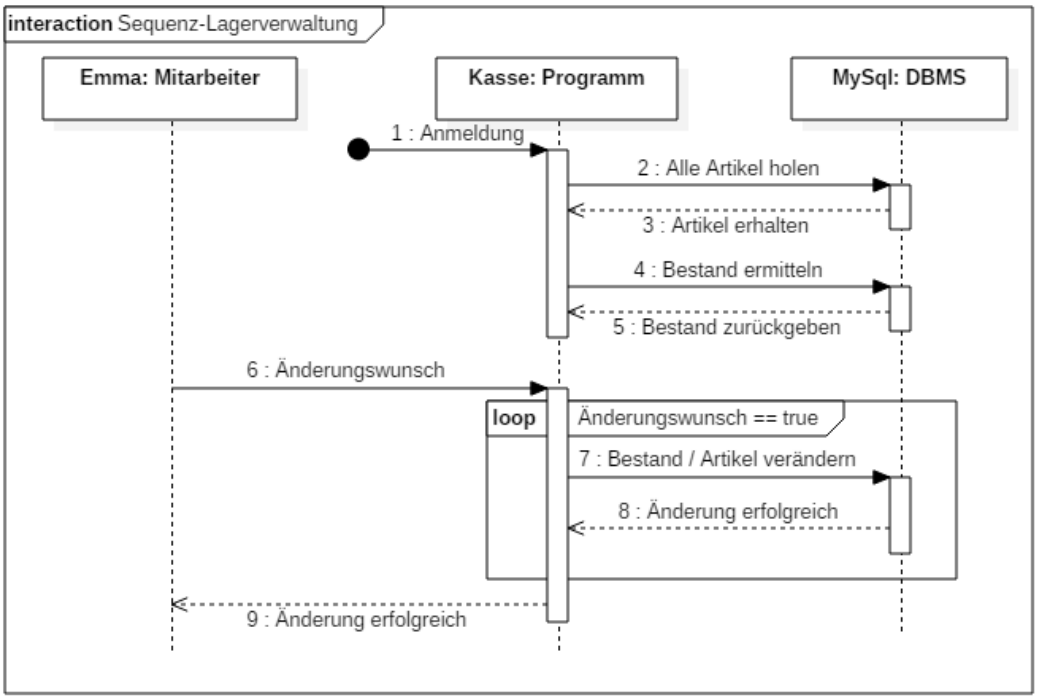
Das Folgende Diagramm zeigt Tante Emma, als authentifiziertes Kassenpersonal. Der Anruf eines Kunden wird abgehandelt und die Gewünschten Artikel in den Warenkorb gelegt. Unter Zuführung der Kundendaten und der Option einer zyklischen Lieferung geht ein Auftrag hervor.



III. Verwaltung

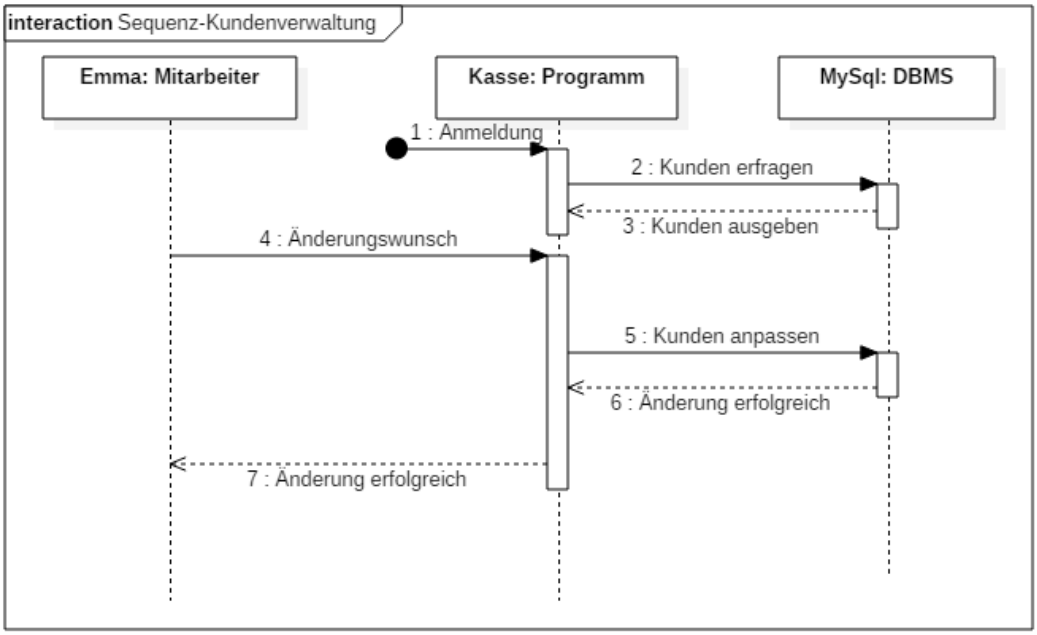
Lagerverwaltung

Der Lagerbestand wird beim Anlegen von Warenkörben automatisch ermittelt, sodass eine Bestellung nicht die Menge der Artikel im Lager übersteigen kann. Dazu wird der Lagerbestand hündisch vom Personal gepflegt. Das folgende Diagramm beschreibt die Mitarbeiterinteraktion und das Zusammenspiel der Kasse mit der Datenbank.



Kundenverwaltung

Kundendaten können vollständig über die Kasse angepasst werden. Das folgende Diagramm beschreibt die Mitarbeiterinteraktion und das Zusammenspiel der Kasse mit der Datenbank.



Benutzerverwaltung

Mitarbeiter können vollständig angepasst werden. Das folgende Diagramm beschreibt die Mitarbeiterinteraktion sowie das Zusammenspiel der Kasse mit der Datenbank.