Abschlussprüfung Sommer 2023

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

Warenwirtschaftssystem

Warenwirtschaftssystem

Abgabetermin: 26.08.23

Prüfungsbewerber:

Moritz Gramer

Hauptstraße 79

77855 Achern

Ausbildungsbetrieb:

SRH BBW Neckargemünd

Im Spitzerfeld 25

69151 Neckargemünd

Abgabetermin: 26.08.23

Prüfungsbewerber:

Moritz Gramer

Hauptstraße 79

77855 Achern

Ausbildungsbetrieb:

SRH BBW Neckargemünd

Im Spitzerfeld 25

69151 Neckargemünd

Inhalt

[Warenwirtschaftssystem 1](file:///C:\Users\gramermo\Documents\Meine%20Prüfung\Prüfungsdokumentation.docx#_Toc140068231)

[1. Einleitung 2](#_Toc140068232)

[1.1. Projektumfeld 2](#_Toc140068233)

[1.2. Projektbeschreibung 3](#_Toc140068234)

[1.3. Projektziel 3](#_Toc140068235)

[1.4. Projektschnittstellen 3](#_Toc140068236)

[2. Planung- und Analysephasen 3](#_Toc140068237)

[2.1. Ist-Analyse 3](#_Toc140068238)

[2.2. Soll-Konzept 4](#_Toc140068239)

[2.3. Projektphasen 4](#_Toc140068240)

[2.4. Wirtschaftlichkeitsanalyse sowie Amortisationsrechnung 4](#_Toc140068241)

[3. Entwurfsphase 4](#_Toc140068242)

[3.1. Benutzeroberfläche entwerfen 4](#_Toc140068243)

[3.2. Datenbankdesign konzipieren 6](#_Toc140068244)

[3.3. Planung der Geschäftslogik 7](#_Toc140068245)

[4. Implementierungsphase 7](#_Toc140068246)

[4.1. Benutzeroberfläche erstellen 7](#_Toc140068247)

[4.2. Datenbankdesign erstellen 7](#_Toc140068248)

[4.3. Geschäftslogik umsetzen 8](#_Toc140068249)

[5. Kundendokumentation 8](#_Toc140068250)

[6. Anhänge 8](#_Toc140068251)

[7. Glossar 8](#_Toc140068252)

# Einleitung

## Projektumfeld

Die folgende Projektdokumentation wurde im Rahmen des IHK-Abschlussprojektes entwickelt. Der Ausbildungsbetrieb des Autors ist das „Berufsbildungswerk Neckargemünd GmbH“.

Auftraggeber des Projekts ist das fiktive Unternehmen „Elektronik Meier GmbH“.

Das Unternehmen besitzt bereits einen Standort in Karlsruhe und plant einen neuen Standort mit Lager in Heidelberg einzurichten.

## Projektbeschreibung

Das Unternehmen „Elektronik Meier GmbH“ hat uns beauftragt, für Ihr neu gebautes Lager ein Warenwirtschaftssystem zu entwickeln. Das Lager soll als Angelpunkt für einen lokalen Verkaufsstandort und für den Online-Versandhandel dienen. Das Unternehmen verkauft hauptsächlich Elektronikartikel und bietet Dienstleistungen wie z.B. Reparaturen an. Die Dienstleistungen werden im Warenwirtschaftssystem nicht berücksichtigt.

## Projektziel

Ziel des Projektes ist die Vereinfachung und Automatisierung des Lagers damit die Lagerarbeiter mit wenig Aufwand und ohne IT-Kenntnisse das Programm intuitiv bedienen können. Wichtig ist es auch für den Kunden die vorhandene Lagerstruktur mit Regalen und Fächern beizubehalten.

Der Kunde strebt dabei eine kostengünstige Variante an.

## Projektschnittstellen

Der Kunde wünscht eine Umsetzung in [C#](#Glossar_C) und eine [SQL](#Glossar_SQL) Datenbank. Die Datenbank wird lokal auf einem Server in der IT-Abteilung beim Kunden implementiert werden.

Wie bereits im [Projektziel](#_Projektziel) beschrieben soll das Programm über eine [GUI](#Glossar_Gui) verfügen. Damit die Mitarbeiter leichten Zugang zu dem Programm haben, soll an strategischen Punkten im Lager Computer eingerichtet werden. Die Positionierung, sowie die Auswahl der Computer wird vom Kunden selbst durchgeführt.

Das Betriebssystem der Lagercomputer ist Windows, daher hat sich der Autor dazu entschieden das Projekt im [.NET Framework](#Glossar_NET_Framework) umzusetzen.

Die Anwendung wird in WPF entwickelt und für die Verbindungen zur SQL Datenbank wird die Package MYSQL.DATA eingebunden. Diese Erweiterung wird als Schnittstelle zwischen dem Datenbankserver und der Anwendung verwendet.

Besonders betroffen von dem Projekt sind die Abteilungen IT und Lager. Ansprechpartner ist der Leiter der IT-Abteilung, Ben Erdrich.

# Planung- und Analysephasen

## Ist-Analyse

An dem neuen Unternehmensstandort ist bereits ein Lager vorhanden. Das Lager ist in verschiedene Regale unterteilt und in jedem Regal befinden sich Fächer für die Verkaufsprodukte. Das Unternehmen setzt auf dynamische Lagerhaltung, das bedeutet die Größe der Fächer kann durch physische Barrieren angepasst werden. Das ermöglicht eine effiziente Nutzung des verfügbaren Platzes und die Möglichkeit

Der neue Unternehmensstandort ist aktuell noch nicht eingerichtet und es befinden sich noch keine Artikel im Lager. Diese werden von den Mitarbeitern in Excel Dateien gespeichert. Sobald das Warenwirtschaftssystem entwickelt und einsatzbereit ist wird das Lager in Betrieb genommen.

Neben dem Lager befindet sich die IT-Abteilung mit einem Raum, der als Serverraum genutzt werden soll.

Die interne IT-Abteilung des Kunden kümmert sich nach Übergabe des fertigen Programms um die Wartung und entwickelt es gegeben falls weiter.

## Soll-Konzept

Angestrebt wird eine einfache Lösung, um den Warenein- und -Ausgang zu dokumentieren und Daten über die gelagerten Waren anzeigen. Die Artikelinformationen sind die Artikelnummer, Preis, Preisaufschlag, Lagerort (Regal und Fach) und das Eingangsdatum.

Des Weiteren soll es die Möglichkeit geben, in der Benutzeroberfläche Waren zu reservieren. Die reservierten Waren sollen direkt als solche zu identifizieren sein.

Die Lagermitarbeiter sollen in der Lage sein Warenein- und Ausgänge in einer grafischen Benutzeroberfläche einzutragen.

Dafür soll eine eigenständige Desktopanwendung entwickelt werden, welche mit einer Datenbank verbunden ist und die gespeicherten Daten abrufen und neue Daten hinzufügen kann.

## Projektphasen

Für die Umsetzung des Projekts stehen dem Autor 80 Stunden zur Verfügung. Zu

Projektbeginn wurden diese in verschiedene Abschnitte unterteilt. Eine grobe Zeitplanung können sie der unteren Tabelle entnehmen.

|  |  |
| --- | --- |
| Projektphase | Geplante Zeit |
| Planungs- und Analysephasen | 7h |
| Entwurfsphase | 17h |
| Implementierungsphase | 29h |
| Dokumentation | 27h |
| Gesamt | 80h |

## Ressourcenplanung

In der folgenden Abbildung sind die benötigen Ressourcen, die für die Durchführung des Projekts nötig sind aufgelistet. Es wird unterschieden zwischen Hardware und Software.

**Hardware**

* Büroarbeitsplatz mit Tower-PC

**Software**

* Windows 10 - Betriebssystem
* Visual Studio 2022 – Entwicklungsumgebung
* GitHub – Versionsverwaltung
* mySQL - Datenbanksystem
* XAMPP mit PHPmyAdmin – Werkzeuge für das lokale Arbeiten mit einem Server
* Microsoft Word – Textverarbeitungsprogramm

## Projektkosten

Die Kosten für das Projekt setzten sich aus Personal- und Ressourcenkosten zusammen.

Der Arbeitsaufwand errechnet sich durch die Anzahl der Stunden \* Stundenlohn. Die Stundenkosten von 70€ wurden durch die Finanzabteilung übermittelt. Die Zeit für die Umsetzung des Projekts beträgt 80 Stunden. Dadurch kommt folgende Rechnung zustande.

**Arbeitsaufwand** = Stundenlohn \* Stundenanzahl

**Arbeitsaufwand** = 70€ \* 80h

**Arbeitsaufwand** = 5600€

Neben den Kosten für den Arbeitsaufwand müssen noch Lizenzkosten beachtet werden. Die Lizenzkosten für den Zeitraum des Projekts entstehen Einmalkosten in Höhe von 350€. Diese Kosten fallen in die Kategorie Ressourcenkosten.

Die Gesamtkosten berechnen sich durch Arbeitsaufwand + Lizenzkosten. Daher entsteht folgender Gesamtpreis:

**Gesamtpreis** = Arbeitsaufwand + Lizenzkosten

**Gesamtpreis** = 5600€ + 350€  
**Gesamtpreis** = 5950€

Der oben ausgerechnete Gesamtpreis ist in dieser Form der Brutto-Gesamtpreis. Um den Netto Gesamtpreis auszurechnen muss der Betrag mit 1,19 multipliziert werden.

**Netto Gesamtpreis** =5950€ \* 1,19

**Netto Gesamtpreis**= 7080,50€

## Make-or-Buy-Entscheidung

Es existieren bereits fertige Lösungen für die gewünschte Anwendung des Kunden. Diese sind in der Regel mit viel mehr Funktionalität ausgestattet und daher wesentlich teurer.

Der Kunde bevorzugt eine Anwendung zu kaufen und selbst weiter daran zu arbeiten, anstatt monatlich für die Software zu zahlen. Flexibilität und Anpassungsfähigkeit spielen bei dem Projekt eine große Rolle.

## Wirtschaftlichkeitsanalyse sowie Amortisationsrechnung

Aufgrund der hohen Kosten, welche für ein fertiges Warenwirtschaftssystem anfallen hat sich der Kunde dazu entschieden, ein schlankes System entwickeln zu lassen, welches auf ihre Bedürfnisse angepasst und wenn, gewünscht von der eigenen IT-Abteilung weiter entwickelt werden kann.

Folgende Tabelle zeigt die geschätzte Einsparung der Arbeitszeit durch das Projekt. Die Mitarbeiterarbeitszeit errechnet sich durch Mitarbeiter \* Arbeitszeit. Das entspricht bei 15 Mitarbeitern mit jeweils 8 Stunden Arbeitszeit 120 Arbeitsstunden.

|  |  |  |
| --- | --- | --- |
| Arbeitszeiteinsparung | Mitarbeiterarbeitszeit  In Stunden | Arbeitszeit |
| 1% | 120 | 1,2h / Tag |
| 2% | 120 | 2,4h / Tag |
| 5% | 120 | 6h / Tag |

Summe der Einsparungen

|  |  |  |  |
| --- | --- | --- | --- |
| Arbeitszeiteinsparung | Arbeitseinsparung | Mitarbeiter Gehalt | Einsparung pro Arbeitstag |
| 1% | 1,2h / Tag | 80€ | 96,00€ |
| 2% | 2,4h / Tag | 80€ | 192,00€ |
| 5% | 6h / Tag | 80€ | 480,00€ |

Amortisationsdauer

|  |  |  |  |
| --- | --- | --- | --- |
| Arbeitszeiteinsparung | Gesamtkosten des Projekts | Einsparung pro Arbeitstag | Amortisationsdauer |
| 1% | 7080,50€ | 96,00€ | ~74 Tage |
| 2% | 192,00€ | ~37 Tage |
| 5% | 480,00€ | ~15 Tage |

Die Gesamtkosten des Projektes wurden in dem Abschnitt Projektkosten ermittelt und dargelegt.

# Entwurfsphase

## Zielplattform

Das Programm soll wie bereits im Soll-Konzept beschrieben als eigenständige Desktopanwendung mit eigener Datenbank umgesetzt werden. Als Datenbank wird mySQL zum Einsatz kommen.

Die Auswahl der Programmiersprache wurde bereits vom Kunden durchgeführt und dieser hat sich für C# entschieden. Die eigene IT-Abteilung benutzt hauptsächlich C# und

Die Computer, die im Lager zum Einsatz kommen laufen mit dem Betriebssystem Windows, daher hat sich der Autor wie im Punkt Projektschnittstellen beschrieben für eine Umsetzung mit WPF entschieden. WPF ist ein Grafik Framework aus dem .Net Framework von Microsoft. Aufgrund der guten Komptabilität mit dem Windows Betriebssystem und der Möglichkeit zur leichten Erstellung von Benutzeroberflächen ist die Wahl der Plattform auf WPF gefallen.

Für die Auswahl der Plattform wurde eine Entscheidung zwischen Windows Forms und WPF gefällt. Dafür wurden die Vorteile beider Frameworks aufgeführt.

|  |  |
| --- | --- |
| **Vorteile WPF** | **Vorteile Windows Forms** |
|  |  |
| Neuer und mehr an aktuelle Standards angelehnt | Älter und besser getestet |
| Flexibler und anpassbarer | Mehr Dokumentation und Beispielmaterial verfügbar |
| Aufteilung zwischen GUI Teil und Code-Teil |  |
| Bessere Performance |  |
| Möglichkeit für Web- und Desktopanwendungen |  |

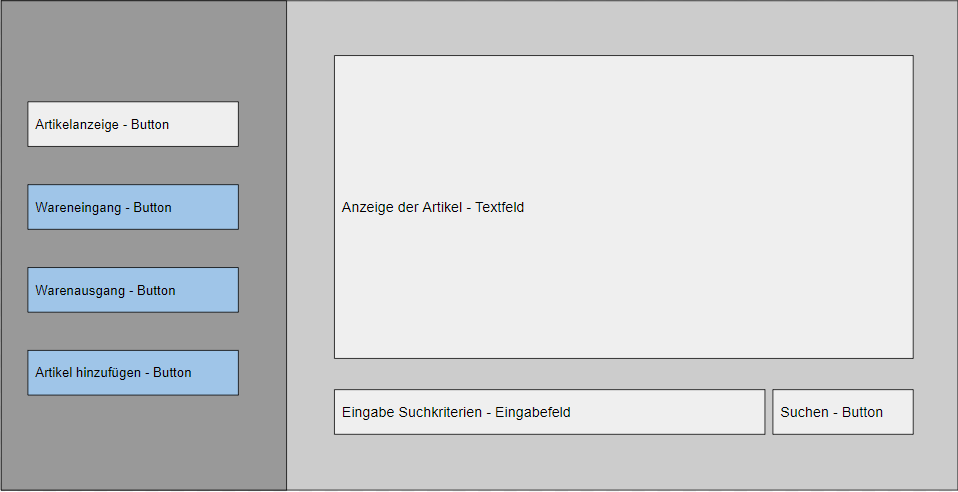
Besonders für WPF spricht die Flexibilität und die Performance. Dazu kommt, dass der Autor selbst mit beide Systemen schon gearbeitet hat und WPF als Wesentlich angenehmer und effizienter wahrgenommen hat.

## Benutzeroberfläche entwerfen

Die Benutzeroberfläche wird aus einem Hauptfenster mit mehreren Unterseiten aufgebaut. Diese Elemente stehen in WPF zur Verfügung. Ein Fenster ist das übergeordnete Element und kann beliebig viele Seite anzeige. Für jede Funktionalität des Programms wird eine eigene Seite erstellt. Für das Einsehen und Suchen von eingelagerten Artikeln, das Anlegen neuer Artikel und den Warenein- und -Ausgang wird ein eigenes Fenster entworfen. Durch ein Menü, welches sich auf der linken Seite befindet kann zwischen Funktionalitäten navigiert werden.

[WPF](#Glossar_WPF) bildet in dieser Hinsicht einen großen Vorteil für das Projekt. [WPF](#Glossar_WPF) ist von der Struktur geteilt zwischen [Frontend](#Glossar_Frontend) und [Backend](#Glossar_Backend). Das ermöglicht es ohne großen Aufwand das Design und die GUI anzupassen ohne die Funktionalität im [Backend](#Glossar_Backend) zu verändern. Diese Struktur wird auch als „Code Behind“ bezeichnet.

Auf der folgenden [Abbildung](#Gui_Artikelanzeige) ist eine grobe Skizze einer Funktionalitätsseite zu sehen. Auf dieser Seite können die verschiedenen Artikel, welche sich im Lager befinden angezeigt werden. Am unteren Bildrand befinden sich Eingabefelder in die Suchkriterien eingegeben werde können nach diesen in der Datenbank gesucht wird. Suchkriterien wird die Artikelnummer. In dem Textfeld werden die Ergebnisse angezeigt, welche in der Datenbank anhand der Suchkriterien gefunden worden sind. Die Planungsskizzen der weiteren Funktionalitäten befinden sich im [Anhang](#_Anhänge).



GUI Abbildung 1 - Grobe Skizze der Artikelanzeige

Im linken unteren Bildrand wird ein Label eingefügt, welches dem Benutzer visuelle Rückmeldung über erfolgreiche Ausführungen oder Fehler bietet. Werden Vorgänge erfolgreich ausgeführt wird in grüner Schrift eine Erfolgsmeldung ausgegeben und bei Fehlernvorgängen wird der Benutzer auf mögliche Fehler hingewiesen. Beispiel für eine mögliche Fehlermeldung ist das nicht Vorhandensein einer Verbindung zur Datenbank.

## Datenbankdesign konzipieren

Den vorläufigen Aufbau der Datenbank können sie der Abbildung Datenbankdesign entnehmen. Die Datenbank wird in die Tabelle Artikel und Lager eingeteilt. Der Autor hat sich dazu entschieden die Tabelle Lager einzufügen, welche die Attribute„Lager\_id“ und Standort des Lagers beinhaltet. Dadurch ist es ohne Probleme möglich die Anwendung auf einem remote Server zu speichern und die Lager für mehrere Standorte zu verwalten.

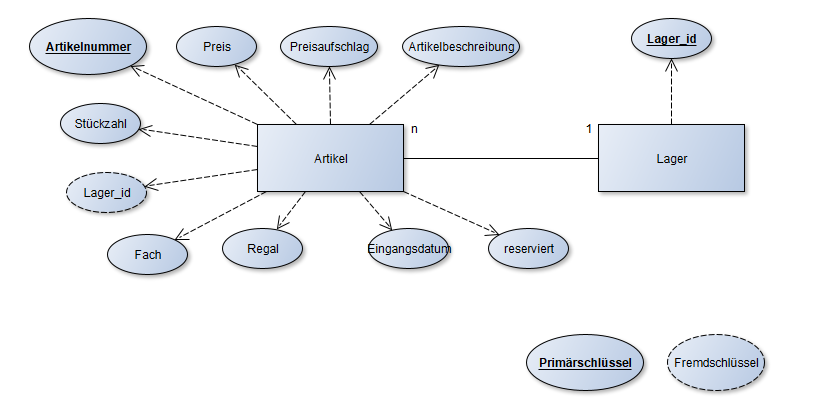


Abbildung Datenbankdesign

Die Datenbanktabelle ‚Artikel‘ erhält alle Informationen über die gespeicherte Ware.

Das Relationenmodell der geplanten Datenbank sieht folgendermaßen aus.

**Lager** (Lager\_id, Lagerstandort)

**Artikel** (Artikelnummer, Artikelbeschreibung, Preis, Lagerbestand, Preisaufschlag, *Lager\_id*, Regal, Fach, Eingangsdatum, reserviert)

## Planung der Geschäftslogik

Die Geschäftslogik kann in vier kleinere Teile unterteilt werden. Es wird Logik benötigt für das Anzeigen der Artikel aus der Datenbank, um den Waren- ein und -Ausgang durchzuführen und um einen neuen Artikel anzulegen. Zum Teil überschneidet sich die zu programmierende Logik. So braucht sowohl der Warenein-, als auch Warenausgang die Möglichkeit den aktuellen Lagerbestand des Artikels im Lager einzusehen und verändern.

Damit ein Artikel im Lager gespeichert und später einfach der Warenein- und -Ausgang verwaltet werden kann muss jeder neue Artikel zuerst angelegt werden. Dafür werden in der GUI in Textfelder erstellt und die zu speichernden Daten werden in der Datenbank gespeichert.

Für die Vorgänge des Programms müssen zuerst die entsprechenden Benutzereingaben ausgelesen werden. Je nach ausgewählter Funktionalität können sich die nötigen Benutzereingaben variieren.

**Anzeigen der Artikel aus der Datenbank**

Damit ein Artikel angezeigt werden kann wird die Artikelnummer vom Benutzer eingegeben. Danach wird eine Anfrage an den Server geschickt und die Ergebnisse werden angezeigt. Falls sich im Lager ein Artikel mit der eingegebenen Artikelnummer befindet werden die Artikelinformationen aus der Datenbank ausgelesen und angezeigt. Bei Eingabe eines Artikels der nicht gefunden wird, wird Benutzer darüber in Kenntnis gesetzt.

Der Vorgang wird hier in diesem Programmablaufplan bildlich beschrieben.

**Wareneingang und Warenausgang**

Die Geschäftslogik des Warenein- und -Ausgangs ähneln sich größtenteils.

Bei beiden Vorgängen wird zuerst die Eingabe für die Artikelnummer und der Lagerbestand ausgelesen. Dann werden bei dem Wareneingang eingegebene Stückzahlen zu dem vorhandenen Lagerbestand in der Datenbank hinzugefügt.

Bei dem Warenausgang wird die eingegeben Stückzahl von dem vorhandenen Lagerbestand auf der Datenbank abgezogen. Dabei muss beachtet werden, dass die eingegebene Stückzahl nicht größer sein darf, als der tatsächlich vorhandene Lagerbestand in der Datenbank. Es kann nicht mehr Ware ausgebucht werden, als im Lager vorhanden ist.

**Artikel hinzufügen**

Um einen Artikel in die Datenbank hinzuzufügen und somit in das Sortiment aufnehmen zu können müssen erst einige Informationen von dem Benutzer eingegeben werden. Diese Informationen sind die Artikelnummer, eine Artikelbeschreibung, der Lagerbestand des Artikels, der Preis, der Preiszuschlag, der Lagerstandort bestehend aus Regalplatz und Fach und die Angabe, ob der Artikel reserviert ist. Das Eingangsdatum des Artikels soll durch SQL direkt eingetragen werden und muss nicht vom Benutzer eingegeben werden.

Neben den Anfragen an die Datenbank prüft die Geschäftslogik die Benutzereingaben und sendet entsprechende Meldungen an den Benutzer des Programms. Beispiel für eine Überprüfung ist, ob die eingegebene Stückzahl wirklich eine Zahl ist. Durch solche Abfragen können Datenbankabstürze verhindert werden.

# Implementierungsphase

## Benutzeroberfläche implementieren

Die Benutzeroberfläche wird in WPF mithilfe der Drag and Drop Funktion erstellt. Zuerst wurde das Hauptfenster eingerichtet. Bei dem Erstellen einer Seite oder eines Fensters werden zwei Dateien automatisch erstellt. Bei dem Fenster „MainWindow“ werden die Dateien „MainWindow.xaml“ und „MainWindow.xamls.cs“ erstellt.

In der Datei mit der Endung „.xaml.cs“ wird die Backend Funktionalität in C# Quellcode gespeichert und in der „.xaml“ Datei befinden sich die Informationen, die benötigt werden um die GUI anzuzeigen.

Beim Start der Anwendung wird die MainWindow.xaml ausgeführt.

Auf der Hauptseite wird ein Frame eingefügt in dem die ausgewählt Funktionalität angezeigt wird. So kann dynamisch und schnell zwischen den einzelnen Funktionalitäten gewechselt werden. Der folgende Code-Schnipsel ist xaml des Frames.

<Frame x:Name="frame\_Content" Margin="165,20,25,20" NavigationUIVisibility="Hidden"></Frame>

Das Feld wird mit einem Namen versehen. Anhand dieses Namens kann das Element im Back-End angesprochen und verändert werden.

Die folgende Abbildung zeigt die Hauptseite mit dem Frame „frame\_Content“ (blauer Rahmen).

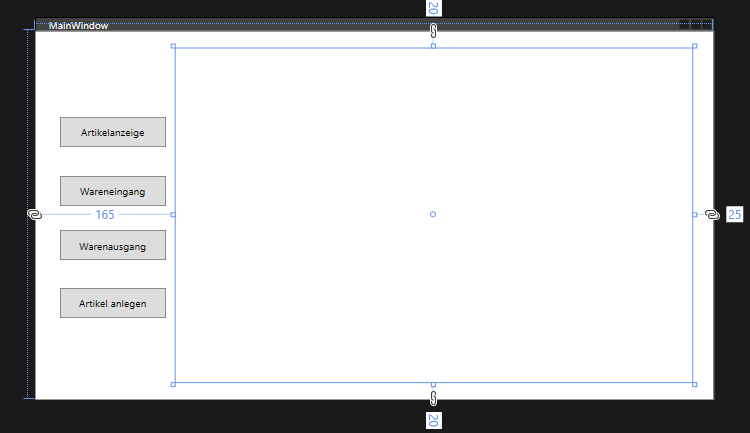
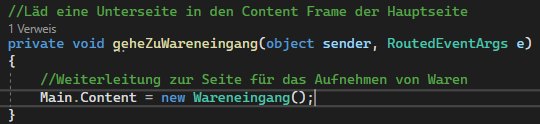


Abbildung 1 Benutzeroberfläche mit Content Feld

In dieses Feld kann im Code eine der Funktionalitäten geladen werden. Dafür wurden vier sogenannte Seiten erstellt. Die Seite ist eine eigene GUI, welche mit den WPF Drag and Drop Funktionen nach Belieben geformt werden kann.

Zuerst wurde die Seite Wareneingang der Skizze entsprechend umgesetzt.

In der folgenden Abbildung ist ein Quellcode-Ausschnitt zu sehen, welcher die GUI des Wareneingangs in das Content Feld lädt.



Bei dem Wareneingang können neue Artikel in das Lager aufgenommen werden. Dafür wurde in der GUI zwei Textfelder erstellt. Ein Textfeld für die Eingabe der Artikelnummer und ein Feld für die Eingabe der Stückzahl von Waren, die in das Lager aufgenommen werden sollen. Darüber hinaus wird ein Button erstellt, welcher den Vorgang abschließt.

Das Erstellen der Fehlermeldungen hat den Autor zunächst vor ein Problem gestellt. Jede einzelne Seite soll die Möglichkeit haben Fehlermeldungen und Erfolgsmeldungen als Rückmeldung für den Benutzer anzuzeigen. Es ist unpraktisch dafür auf jedem Contentfeld eine eigene Ausgabe einzurichten. Die Ausgabe der Rückmeldungen soll daher in der übergeordneten Seite angezeigt werden, welches das Contentfeld lädt. Die Contentfelder mit den Funktionalitäten haben nicht direkt Zugriff auf die GUI des übergeordneten Fensters. Dadurch muss erst eine Möglichkeit gefunden werden die Methoden des Fensters aufzurufen. Das Problem wurde durch die folgende Zeile Code gelöst. Das Label kann angesprochen werden nachdem das aktuelle Hauptfenster in einer Variable vom Typ Window gespeichert wird.

//das Hauptfenster(MainWindow.xaml.cs) wird in eine Variable gespeichert

Window window = Application.Current.MainWindow;

Bei der Methode zum Hinzufügen der Artikel ins Lager wurde ein Bug gefunden. Wurde bei dem Eingabefeld für die Artikelnummer einfach Artikelnummer eingetragen führte das dazu, dass bei der Datenbank jeder Artikel ausgewählt wurde. Die folgende Abbildung ist eine Verdeutlichung, wie es zu diesem Bug kommt



Bei dem Befehl, welcher auf der Datenbank ausgeführt wurde befand sich in der ‚Where‘ Klausel eine Abfrage, die bei jedem Artikel wahr ist. Dadurch kam es zu unerwünschten Veränderungen in der Datenbank. Um Das Problem zu lösen wurde in der Klasse ‚Datenbankverbindung‘ eine Abfrage eingebaut, welche überprüft, ob es sich bei der Benutzereingabe um ‚Artikelnummer‘ handelt. Ist dies der Fall wird ein leerer String zurückgesendet.

## Datenbank implementieren

Um auf die Datenbank zugreifen zu können wird ein Benutzer mit Passwort und Rechten für die Datenbank benötigt. Dieser wurde mit folgendem Befehl erstellt.

CREATE USER 'Lagerarbseiter'@'localhost' IDENTIFIED BY 'tempPassword';

Das Passwort ist nur ein vorübergehender Platzhalter, welcher vom Kunden selbst durch ein starkes Passwort ersetzt werden muss. Für die Auswahl des Passworts sollte sich an aktuelle Standards gehalten werden.

Der Benutzer benötigt Rechte um auf die Datenbank zuzugreifen und Daten zu verändern. Mit diesem Befehl werden dem angelegten Nutzer die nötigen Privilegien dafür gegeben.

GRANT ALL PRIVILEGES ON lagerdatenbank . \* TO 'Lagerarbeiter'@'localhost';

Zuerst wurde die Tabelle Lager erstellt, da die Tabelle Artikel einen Verweis auf die Tabelle Lager benötigt. Die Datenbanktabellen wurden mit den folgenden Befehlen in phpMyAdmin erstellt.

SQL Befehl, um die Tabelle Lager zu erstellen.

[Create](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) [Table](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/create-table.html) Lager (

 Lager\_ID [int](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/numeric-types.html) [NOT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/logical-operators.html#operator_not) NULL,

 standort [varchar](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-types.html)(255),

 PRIMARY KEY(Lager\_ID)

);

SQL Befehl, um die Tabelle Artikel zu erstellen.

CREATE TABLE Artikel (

Artikelnummer varchar(10) NOT NULL,

Artikelbeschreibung varchar(255),

Preis varchar(10),

Lagerbestand int,

Preisaufschlag varchar(10),

Lager\_id int,

Regal int,

Fach int,

Datum datetime,

reserviert varchar(4) DEFAULT 'Nein',

PRIMARY KEY(Artikelnummer),

FOREIGN KEY (lager\_id) REFERENCES lager(lager\_ID)

);

Um einen Artikel aus dem Lager anzuzeigen muss die Artikelnummer aus dem Eingabefeld ausgelesen werden und eine SELECT Anfrage nach dieser Artikelnummer an die Datenbank durchgeführt werden.



SQL SELECT

Das ist ein Beispielbefehl wie Daten von der Datenbank ausgelesen werden können. In diesem Fall werden alle Informationen über den Artikel mit der Artikelnummer ‚1234‘ geladen. Artikelnummer ist in der Tabelle Artikel der Primärschlüssel, dadurch kann pro Anfrage nur ein Artikel gefunden werden. Die Artikelnummer in der Datenbank kann aus Zahlen und Buchstaben zusammengesetzt sein und vom Kunden frei gewählt werden.

Bei dem Erstellen der Datenbankverbindung traten erstmals Probleme auf. Durch die Unterteilung in verschiedene Seiten musste von unterschiedlichen Teilen des Codes die Möglichkeit bestehen auf die Datenbank zuzugreifen. Der Autor versuchte dieses Problem mit einer statischen Klasse zu lösen, was sich nach gründlicher Überlegung als nicht sinnvoll herausgestellt hat. Die statische Klasse würde beim Start des Programms eine Verbindung zur Datenbank erstellen mit der die einzelnen Seiten Abfragen und Befehle auf der Datenbank ausführen konnten. Das stellte sich als sehr schlechte Methode heraus und brachte allerlei Probleme mit sich. Es wäre so nicht möglich mit mehreren Computern auf die Datenbank zuzugreifen und es ist generell nicht sinnvoll eine Datenbankverbindung aufzubauen und nicht mehr zu schließen.

Danach wurde eine dynamischere Methode der Datenbankverbindung angestrebt. Die Datenbankverbindung sollte in einer eigenen Klasse erfolgen, jedoch sollte jede Abfrage eine eigene Instanz der Datenbankverbindung erstellen, welche sich nach erfolgreichem Durchführen oder nach Fehlern wie z.B. einer ungültigen Benutzereingabe direkt schließt.

Dafür wurde die Klasse ‚Datenbankverbindung‘ erstellt. In dieser Klasse werden sämtliche Vorgänge, welche mit der Datenbank zusammenhängen bearbeitet. Die Klasse regelt sämtliche Datenbankanfragen. Sie hat eine private Variable vom Typ „MySQLConnection“ und den privaten String „connectionString“, in dem alle Daten gespeichert sind welche für die Datenbankverbindung nötig sind. Der String sieht folgendermaßen aus.

private string connectionString = "server=localhost;user=root;database=lagerdatenbank;port=3306;password=";

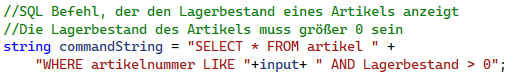
Zur Qualitätssicherung wurde die Methode ‚überprüfeVerbindung‘ erstellt. Bevor eine Anfrage an die Datenbank gesendet wird prüft diese Methode ob eine Verbindung besteht und liefert „true“ bei aktiver Verbindung oder „false“, wenn keine Verbindung besteht zurück. Bei Rückgabewert „false“ und damit keiner aktiven Verbindung zur Datenbank werden keine SQL Befehle ausgeführt. Der Benutzer wird mit einer Fehlermeldung über das Fehlen der Verbindung hingewiesen.

## Geschäftslogik umsetzen

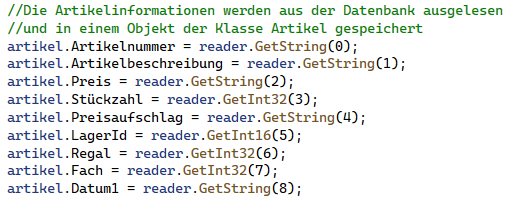
Die Geschäftslogik wurde Schritt für Schritt umgesetzt. Zuerst wurde die Funktion der Anzeige eines Artikels aus dem Lager programmiert. Nachdem die GUI wie in

In der Abbildung (einfügen) ist der Vorgang zum Anzeigen eines Artikels als Programmablaufplan zu sehen.

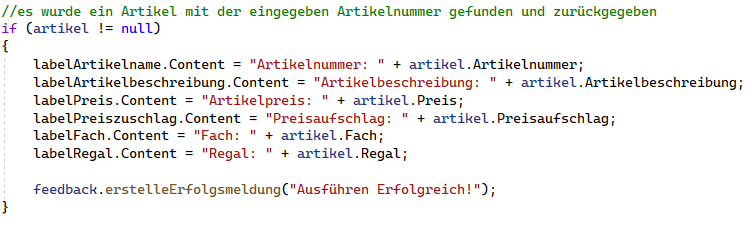
Der folgende SQL Befehl liefert die Artikelinformationen von der Datenbank.



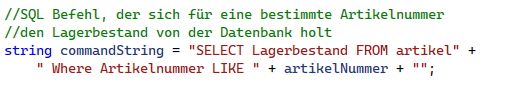
Der String wird in einen „MySqlCommand“ umgewandelt und kann in dieser Form von einem „MySqlDataReader“ ausgeführt werden. Die Daten, welche der SELECT Befehl zurückgibt, werden dann in einer Instanz der Klasse Artikel gespeichert. Die Klasse Artikel verfügt besitzt eine private Variable für jede Information, die in der Datenbank über einen Artikel gespeichert wird. Mit „gettern“ und „settern“ können die Variable befüllt und Informationen weitergegeben werden.



Nachdem das Objekt erfolgreich mit Daten bestückt wurde, wird die Benutzeroberfläche aktualisiert und die Artikelinformationen sowie eine Rückmeldung für den Benutzer wird angezeigt.



Im Anschluss wurden wegen der Ähnlichkeit der Anforderungen die Funktionalitäten des Warenein- und -Ausgangs entwickelt. Der erste Schritt bestand aus dem Erstellen einer Methode in der Klasse Datenbankverbindung, welche für eine eingegebene Artikelnummer den Lagerbestand des Artikels aus der Datenbank ermittelt. Die Methode führt einen SQL Select Befehl durch

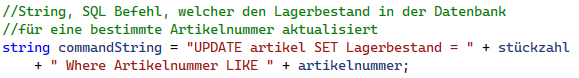


Der Lagerbestand wird and die Aufrufklasse gegeben. Falls kein Lagerbestand ermittelt werden konnte wird ein leerer „String“ zurückgegeben. Die Klasse, welche die Methode aufgerufen hat gibt dann eine Fehlermeldung in der GUI aus.

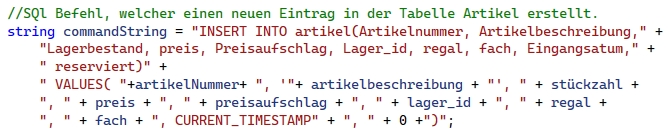
Der Wareneingang addiert die eingegeben Stückzahl mit dem vorhanden ermittelten Lagerbestand und der Warenausgang subtrahiert die Stückzahl vom Lagerbestand, dies wird nur durchgeführt, wenn die eingegeben Stückzahl kleiner als der Lagerbestand ist.

Danach wird der Lagerbestand in der Datenbank aktualisiert. Dafür wird die Methode

„setztNeuenLagerbestandFürArtikelnummer“ in der Datenbankklasse aufgerufen. Die Methode nimmt als Parameter, den neuen Lagerbestand und die Artikelnummer und führt einen SQL Update Befehl durch.



Für die Funktionalität des Anlegens eines Artikels werden Artikelinformationen von dem Benutzer eingetragen, welche in ein SQL Insert Into Befehl eingetragen werden. Dieser Befehl erstellt einen neuen Eintrag in der Tabelle „artikel“ in der Datenbank.



Es ist zu beachten, dass das Datum nicht vom Benutzer eingegeben werden muss. Mit der in SQL integrierten Funktion „CURRENT\_TIMESTAMP“ wird das aktuelle Datum mit Uhrzeit abgefragt und als Attribut Eingangsdatum in die Datenbank eingetragen.

Um den Anwender über mögliche Fehlereingabe und korrekt ausgeführte Vorgänge zu informieren wurde die öffentliche Klasse ‚Feedback‘ erstellt. Eine Instanz der Klasse ermöglicht es Fehlermeldungen und Erfolgsmeldung in der GUI anzuzeigen. Bei Fehlermeldungen wird der Text in roter Farbe ausgegeben und bei Erfolgsmeldungen in grün.

(Code Beispiele von Feedback Klasse einfügen)

# Kundendokumentation

# Anhang

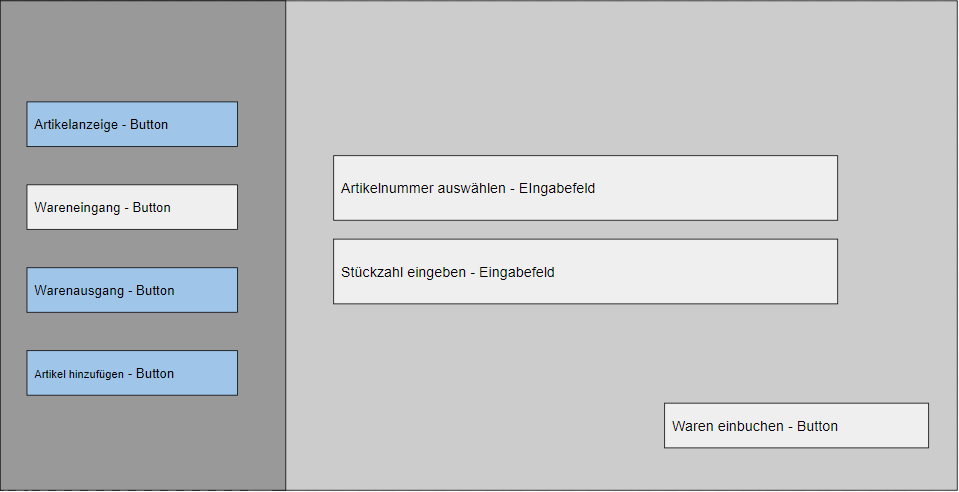


Abbildung 2 - Grobe Skizze des Wareneingangs

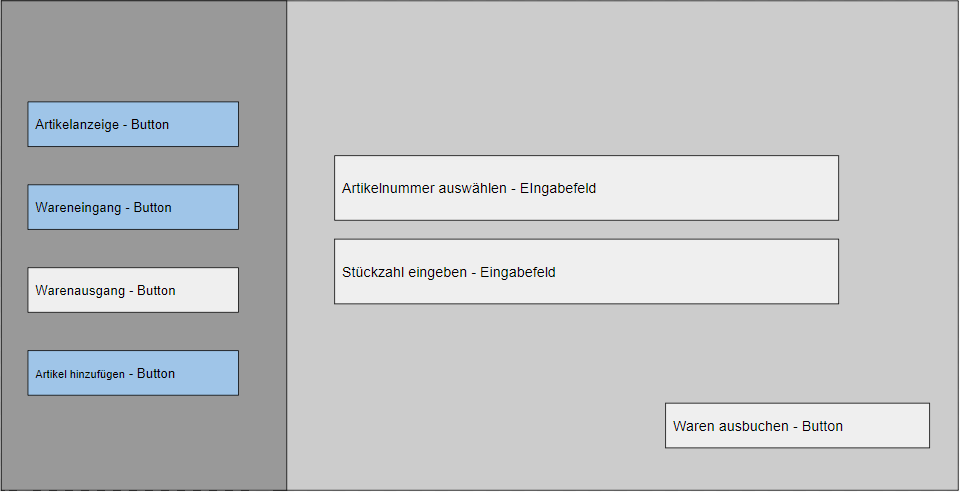


Abbildung 3 - Skizze des Warenausgangs

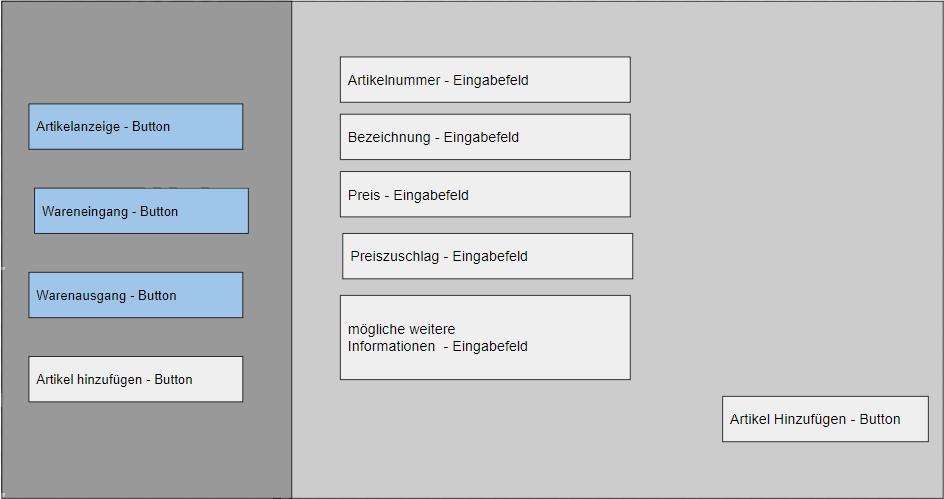


Abbildung 4 - Grobe Skizze Artikel Hinzufügen

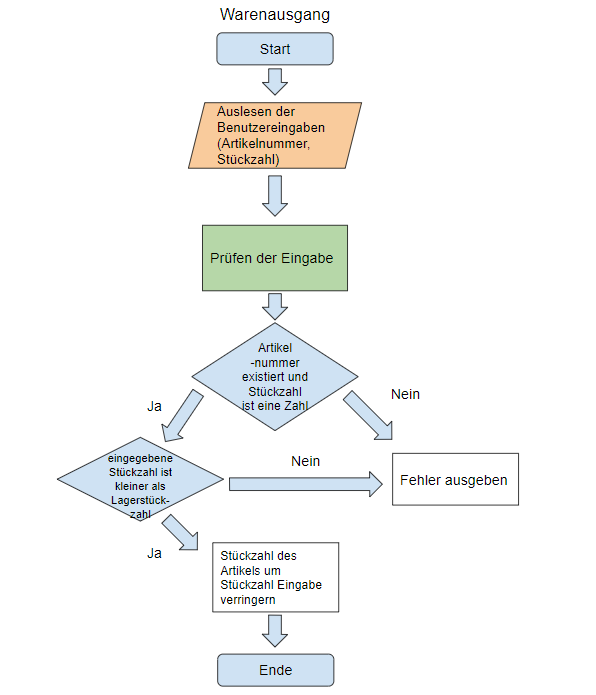


Abbildung 2 Warenausgang PAP

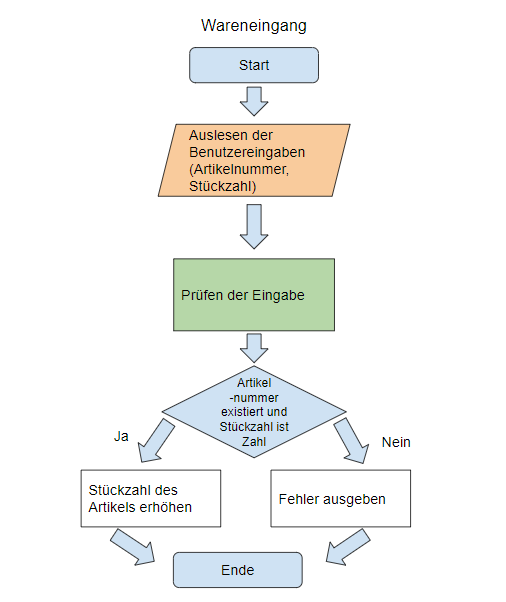


Abbildung 3 Wareneingang PAP

# Glossar

|  |  |
| --- | --- |
| Begriff | Bezeichnung |
| C# | Programmiersprache |
| GUI | Grafische Benutzeroberfläche (Graphical User Interface) |
| SQL | Structured Query Language – Standardsprache für die Erstellung und Bearbeitung von Datenbanken |
| WPF | Windows Presentation Foundation (Tool zur Erstellung von Benutzeroberflächen bei Windows) |
| Framework |  |
| .NET Framework | Technologie, die das Erstellen und Ausführen von Windows-Apps unterstützt |
| Package | Code-Bibliothek, welche erweitere Funktionen bieten |
| Frontend | Teil der Software, welcher der Nutzer sieht und mit dem interagiert werden kann. |
| Backend | Teil der Software, welcher nur im Hintergrund ausgeführt wird und nicht vom Nutzer gesehen werden kann. |
| XAML | extensible application markup language, eine von Microsoft entwickelte Auszeichnungssprache zur strukturierten Beschreibung von Benutzeroberflächen. |
| PHPMyAdmin | Anwendung zur Verwaltung einer mySQL Datenbank |
| MySQL | Datenbank Service |
| Bug | Fehler in Computerprogrammen |
| Frame | Steuerelement in WPF zur Unterstützung der Inhaltsnavigation |
| Primärschlüssel | Der Primärschlüssel kommt in relationalen Datenbanken zum Einsatz und wird zur eindeutigen Identifizierung eines Datensatzes verwendet |
| Fremdschlüssel | Ein Fremdschlüssel ist eine Spalte oder eine Spaltengruppe in einer Tabelle, deren Werte den Werten des Primärschlüssels in einer anderen Tabelle entsprechen |
| Excel | Tabellenkalkulationssoftware |
| Programmablaufplan | Eine grafische Darstellung zur Umsetzung eines Algorithmus in einem Programm und beschreibt die Folge von Operationen zur Lösung einer Aufgabe |
| Fenster | Engl. Window. Ein Programmbaustein in WPF zur Anzeige von Inhalten |
| Seite |  |
| Getter und setter | Ist in der objektorientierten Programmierung eine spezielle Methode, die eine einzelne Eigenschaft (Attribut) eines Objekts abfragt oder ändert |
| String | In der Informatik eine endliche Folge von Zeichen (z. B. Buchstaben, Ziffern, Sonderzeichen und Steuerzeichen) aus einem definierten Zeichensatz |