Abschlussprüfung Sommer 2023

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

# 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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#_Toc141102578)

[1. Einleitung 3](#_Toc141102579)

[1.1. Projektumfeld 3](#_Toc141102580)

[1.2. Projektbeschreibung 3](#_Toc141102581)

[1.3. Projektziel 3](#_Toc141102582)

[1.4. Projektschnittstellen 3](#_Toc141102583)

[2. Planung- und Analysephasen 4](#_Toc141102584)

[2.1. Ist-Analyse 4](#_Toc141102585)

[2.2. Soll-Konzept 4](#_Toc141102586)

[2.3. Projektphasen 4](#_Toc141102587)

[2.4. Ressourcenplanung 5](#_Toc141102588)

[2.5. Projektkosten 5](#_Toc141102589)

[2.6. Make-or-Buy-Entscheidung 5](#_Toc141102590)

[2.7. Wirtschaftlichkeitsanalyse sowie Amortisationsrechnung 6](#_Toc141102591)

[3. Entwurfsphase 7](#_Toc141102592)

[3.1. Zielplattform 7](#_Toc141102593)

[3.2. Benutzeroberfläche entwerfen 7](#_Toc141102594)

[3.3. Datenbankdesign konzipieren 8](#_Toc141102595)

[3.4. Planung der Geschäftslogik 9](#_Toc141102596)

[4. Implementierungsphase 10](#_Toc141102597)

[4.1. Benutzeroberfläche implementieren 10](#_Toc141102598)

[4.2. Datenbank implementieren 11](#_Toc141102599)

[4.3. Geschäftslogik umsetzen 13](#_Toc141102600)

[5. Funktionsüberprüfung / Qualitätskontrolle 16](#_Toc141102601)

[6. Fazit 17](#_Toc141102602)

[6.1. Soll- ist Vergleich 17](#_Toc141102603)

[6.2. Gelerntes 17](#_Toc141102604)

[6.3. Ausblick 17](#_Toc141102605)

[7. Kundendokumentation 17](#_Toc141102606)

[8. Anhang 20](#_Toc141102607)

[9. Quellen 26](#_Toc141102608)

[10. Kundendokumentation 26](#_Toc141102609)

[15. Glossar 29](#_Toc141102610)

[16. Quelltextdokumentation 31](#_Toc141102611)

# 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 für den Kunden ist es, 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 Regale und Fächer kann durch physische Barrieren angepasst werden. Das ermöglicht eine effiziente Nutzung des verfügbaren Platzes und die Möglichkeit für Waren mit hohem Lagerbestand mehr Platz zur Verfügung zu stellen.

Der neue Unternehmensstandort ist aktuell noch nicht eingerichtet und es befinden sich noch keine Artikel im Lager. 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 wartet das Programm nach Übergabe und entwickelt es gegebenenfalls weiter.

## Soll-Konzept

Angestrebt wird eine einfache Lösung, um den Warenein- und -Ausgang zu dokumentieren und Daten über die gelagerten Waren anzuzeigen. 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 identifizierbar 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 | 7 h |
| Entwurfsphase | 17 h |
| Implementierungsphase | 28 h |
| Dokumentation | 27 h |
| Abnahme | 1 h |
| Gesamt | 80 h |

## Ressourcenplanung

In der folgenden Abbildung sind die 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 und 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 mehr Funktionalität ausgestattet und daher wesentlich teurer.

Der Kunde bevorzugt eine Anwendung zu kaufen und selbst weiter daran zu arbeiten, anstatt die Software in einem Abo oder SaaS System zu beziehen. 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 weiterentwickelt 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 verwendet.

Die Auswahl der Programmiersprache wurde bereits vom Kunden vorgegeben. Der Kunde wünscht die Umsetzung in C#.

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 Kompatibilität 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 beiden Systemen schon gearbeitet hat und WPF als Wesentlich angenehmer und effizienter wahrgenommen hat.

## Entwicklungsprozess

Um eine effiziente Herangehensweise an das Projekt zu gewährleisten wurde sich vorab für eine Projektmanagement Methode entschieden. Agilität und schnelle Anpassungen an veränderte Begebenheiten sind zwei Werte, die unbedingt durch das Projektmanagement gefördert werden könne. Für den Entwicklungs- und Planungsprozess wurde auf die Kanban Methode gesetzt. Bei Kanban werden Prozesse in einzelne Punkte unterteilt, welche nach und nach bearbeitet werden.

Um dies in diesem Projekt anzuwenden wurde die gesamte Anwendung in verschieden Funktionalitäten eingeteilt und diese nach und nach abgearbeitet. Es gibt Listen für Aufgaben, die erledigt werden müssen und Aufgaben, welche bereits erledigt sind. Das ermöglicht es Teams effizient zu arbeiten und die zu erledigenden Aufgaben gut untereinander aufzuteilen.

Kanban eignet sich für einzelne Personen, wobei kleine und große Teams von der Einfachheit, der einfachen Koordination und Aufgabenverteilung stark profitieren. Zu dem bietet Kanban eine visuelle Rückmeldung der erledigten und zu erledigenden Aufgaben, welche auf dem Kanban Board ausgehangen werden.

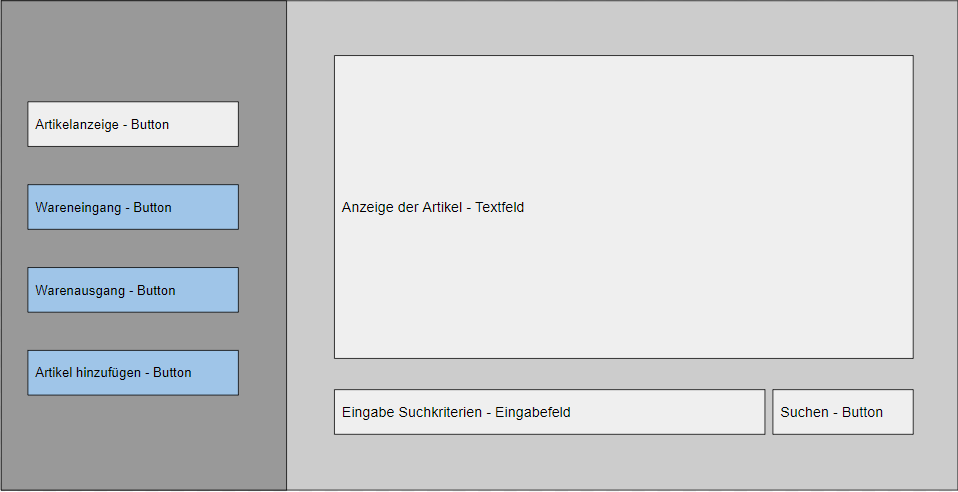
## Benutzeroberfläche entwerfen

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 Seiten anzeigen. 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, denn es ist von der Struktur geteilt zwischen [Frontend](#Glossar_Frontend) und [Backend](#Glossar_Backend). Das ermöglicht es, 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 werden können, nach diesen in der Datenbank gesucht wird. Suchkriterium 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 eine Ausgabe 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 Fehlvorgängen wird der Benutzer auf mögliche Fehler hingewiesen. Beispiel für eine mögliche Fehlermeldung ist das Nichtvorhandensein einer Verbindung zur Datenbank.

Die Seite des Warenein- und Warenausgangs verfügen neben den Menüpunkten im linken Bildrand zwei Eingabefelder. Dort können Artikelnummer und eine Stückzahl eingegeben werden. Die Abbildungen 1 und 2 sind grobe Skizzen des Seitenaufbaus.

Über den Knopf im unteren rechten Bildrand kann der Vorgang des Ein- oder Ausbuchens gestartet werden.

Die Skizze in Abbildung 3 zeigt den Aufbau der Seite, auf der ein neuer Artikel ins Lager aufgenommen werden kann. Diese Seite enthält einige Textfelder, in denen Artikelinformationen eingegeben werden können. Die Seite enthält Eingabefelder für Artikelnummer, Artikelbeschreibung, Preis, Preisaufschlag, Stückzahl, Regal und Fach. Außerdem verfügt die Seite über ein Markierungsfeld. Ist das Häkchen bei dem Feld gesetzt wird der Artikel als reserviert in die Datenbank eingetragen. Standardmäßig soll das Feld ohne aktives Häkchen angezeigt werden.

## Datenbankdesign konzipieren

Den vorläufigen Aufbau der Datenbank können Sie der Abbildung 8 Datenbankdesign entnehmen. Die Abbildung zeigt ein Entity-Relationship-Modell des Datenbankentwurfs. Die Datenbank wird in die Tabellen Artikel und Lager unterteilt. 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 remote auf einem Server zu speichern und die Lager für mehrere Standorte zu verwalten ohne das große Änderungen im Quellcode durchgeführt werden müssen.

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

Das Relationen Modell der geplanten Datenbank sieht folgendermaßen aus.

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

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

Die Artikelnummer wird der Primärschlüssel der Tabelle Artikel. Das ist sinnvoll, da die Artikelnummer einzigartig ist und jeder Eintrag in der Tabelle unbedingt eine Artikelnummer benötigt. Das Attribut ‚Lager\_id‘ ist der Fremdschlüssel und referenziert in die Tabelle Lager.

## 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 Warenein- 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 zu 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 Eingabefelder erstellt und die eingegebenen 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 einer nicht vorhandenen Artikelnummer, wird der Benutzer darüber in Kenntnis gesetzt.

Der Vorgang wird in Anlage 6 in einem 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 können nicht mehr Artikel ausgebucht werden, als im Lager vorhanden sind.

**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 somit 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. Beim 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 in XAML, die benötigt werden, um die GUI anzuzeigen.

Beim Start der Anwendung wird die „MainWindow.xaml“ ausgeführt und läd die GUI.

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 zeigt den xaml Code 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).

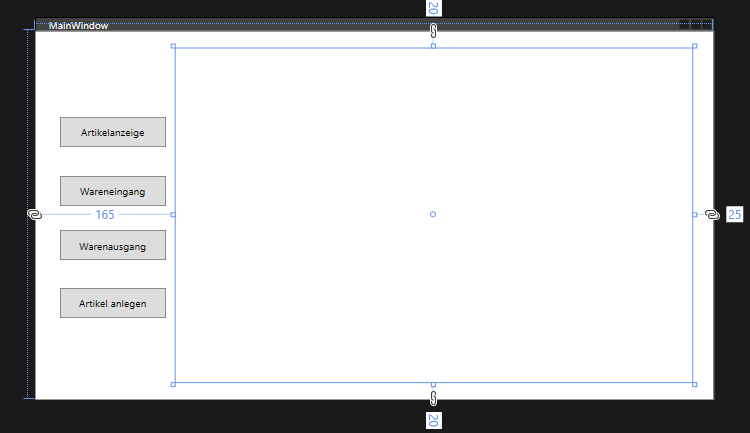
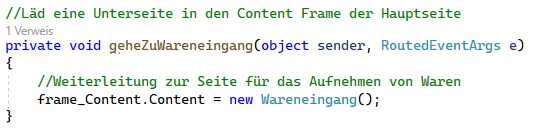


Abbildung 1 Benutzeroberfläche mit Content Feld

In diesem Feld kann im Code eine der Funktionalitäten geladen werden. Dafür wurden vier sogenannte Seiten erstellt. Jede Seite ist eine eigene GUI, welche auch mit den WPF Drag and Drop Funktionen nach Belieben geformt werden kann. Zuerst wurde die Seite des Wareneingangs der Skizze entsprechend umgesetzt. Zum Start des Programms wird standardmäßig die Funktionalität zum Anzeigen eines Artikels geöffnet.

In der folgenden Abbildung ist ein Quellcode-Ausschnitt zu sehen, welcher die GUI des Wareneingangs in das Content Feld lädt. Auf dieselbe Weise werden die anderen Seiten in das Content Feld geladen.



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. Darüber hinaus wird ein Button erstellt, welcher den Vorgang des Einbuchens startet. In die Textfelder wurden Begrenzungen für die Länge der Benutzereingaben eingesetzt. Vorteil dieses Vorgehens ist, dass Fehleingaben schon durch die GUI ausgefiltert werden können, ohne dafür Logik zu programmieren. Es ist einfacher und sicherer solche Fälle direkt abzufangen.

Auf die Seite der Artikelanzeige wurden neun Ausgabefelder hinzugefügt, welche die einzelnen Artikelinformationen anzeigen. Jedes Element erhält einen eigenen Namen womit es vom Backend identifiziert und angesprochen werden kann. Die Seite verfügt außerdem über ein Eingabefeld für die Artikelnummer und einen Button ‚Suchen‘.

Das Anzeigen 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 jeder Seite eine eigene Ausgabe einzurichten. Die Ausgabe der Rückmeldungen soll daher in dem übergeordneten Fenster angezeigt werden, welches die einzelnen Seiten in den Frame lädt. Die Seiten 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;

Die [Abbildungen](#Abbildungen_GUI) 10-14 zeigen das fertige Design der GUI.

## 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" \t "mysql_doc) [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" \t "mysql_doc)(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)

);

Die Abbildung 9 Datenbankdesign zeigt den Aufbau der Datenbank im phpMyAdmin Designer.

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‘ selektiert. 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.

Beim 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 einige 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=Lagermitarbeiter;database=lagerdatenbank;port=3306;password=tempPassword“;

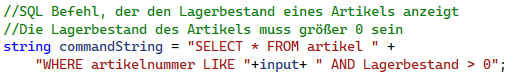
Zur Qualitätssicherung wurde die Methode ‚überprüfeVerbindung‘ erstellt. Bevor eine Anfrage an die Datenbank gesendet wird prüft diese Methode ob eine Verbindung besteht und liefert „true“ bei aktiver Verbindung oder „false“, wenn keine Verbindung besteht zurück. Bei Rückgabewert „false“ und damit keiner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

## Geschäftslogik umsetzen

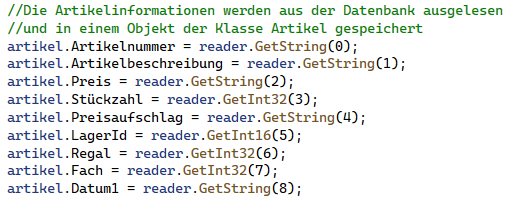
Nachdem GUI und Datenbank erstellt und der Datenbankzugriff funktionierte, wurde die Geschäftslogik umgesetzt. Zuerst wurde die Funktion der Anzeige eines Artikels aus dem Lager programmiert.

In der Abbildung 6 ist der Vorgang zum Anzeigen eines Artikels als Programmablaufplan zu sehen.

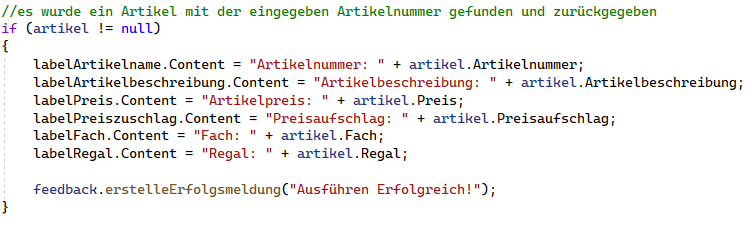
Der folgende SQL Befehl liefert die Artikelinformationen von der Datenbank.



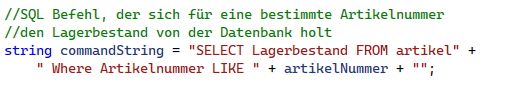
Der String wird in einen „MySqlCommand“ umgewandelt und kann in dieser Form von einem „MySqlDataReader“ ausgeführt werden. Die Daten, welche der SELECT Befehl zurückgibt, werden dann in einer Instanz der Klasse Artikel gespeichert. Die Klasse Artikel verfügt besitzt eine private Variable für jede Information, 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 den Lagerbestand des Artikels mit der eingegebenen Artikelnummer ermittelt.

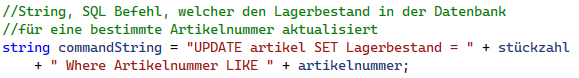


Der Lagerbestand wird an die Aufrufklasse gegeben. Falls kein Lagerbestand ermittelt werden konnte wird ein leerer String zurückgegeben. Die Klasse, welche die Methode aufgerufen hat gibt dann eine Fehlermeldung in der GUI aus.

Der Wareneingang addiert die eingegeben Stückzahl mit dem vorhandenen 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.



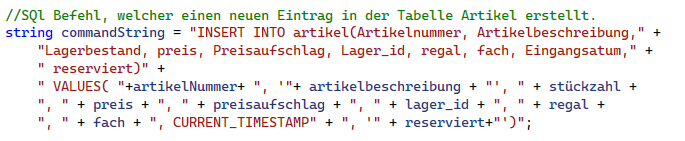
Bei diesem Vorgang zum Aktualisieren der Artikel im Lager wurde ein Bug gefunden. Wurde bei dem Eingabefeld für die Artikelnummer das Wort Artikelnummer eingetragen führte das dazu, dass bei der Datenbank jeder Artikel ausgewählt und der die Erhöhung des Lagerbestands bei allen Artikeln im Lager durchgeführt wurde. Die folgende Abbildung ist eine Verdeutlichung, wie es zu diesem Fehler 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.

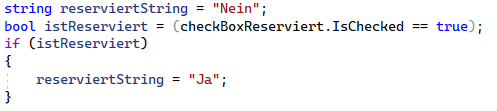
Um Das Problem zu lösen wurde in der Klasse ‚Datenbankverbindung‘ eine Abfrage eingebaut, welche überprüft, ob es sich bei der Benutzereingabe um ‚Artikelnummer‘ handelt. Ist dies der Fall wird ein leerer String zurückgesendet.

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.



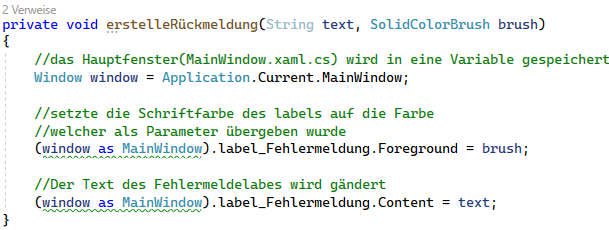
Es ist zu beachten, dass das Datum nicht vom Benutzer eingegeben werden muss. Mit der in SQL integrierten Funktion „CURRENT\_TIMESTAMP“ wird das aktuelle Datum mit Uhrzeit abgefragt und als Attribut Eingangsdatum in die Datenbank eingetragen.

Ein Artikel kann als reserviert markiert werden in dem ein Häkchen bei dem Feld reserviert in der GUI eingetragen wird. Der Status des Feldes wird abgefragt und in der Variable ‚istReserviert‘ gespeichert. Dementsprechend wird dann die Variable verändert, welche auf der Datenbank im Feld Reserviert eingetragen wird.



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.

Der folgende Code-Ausschnitt zeigt die Methode „erstelleRückmeldung“, welche von verschiedenen Stellen im Programm aufgerufen werden kann und eine visuelle Meldung für den Benutzer anzeigt. Die Methode kann als Fehlermeldung oder positive Rückmeldung eingesetzt werden. Je nachdem welche Farbe in dem Übergabeparameter ‚brush‘ übergeben wird.



# Funktionsüberprüfung / Qualitätskontrolle

Um die korrekte Funktionsweise des Programms sicherzustellen hat der Autor sämtliche programmierte Funktionalitäten direkt nach der Umsetzung getestet. Dadurch konnten einige Fehler und zukünftige potentielle Fehlerquellen schnell ausgeschlossen werden. Des Weiteren wurde nach Fertigstellung des Programms weitere Tests durchgeführt, um alles noch einmal abschließen zu testen. Die Testfälle sind nach Funktionalität sortiert und jede wurde einzeln getestet. Jede Funktionalität wurde mit 2 verschiedenen Testfällen getestet.

**Artikel anlegen Testfälle**

**Testfall 1:**

Ein neuer Artikel wird im Lager angelegt. Dafür werden folgende Testwerte in der GUI eingegeben.

Artikelnummer: 1234

Stückzahl: 20

Artikelbeschreibung: Ein Artikel im Lager, der reserviert ist

Preis: 100

Preisaufschlag: 10

Regal: 1

Fach: 1

Reserviert: Ja

**Erwartetes Ergebnis**

Die eingegebenen Informationen werden korrekt ausgelesen und an der richtigen Stelle in die Datenbank eingetragen. Außerdem wird automatisch das aktuelle Datum mit Uhrzeit als Eingangsdatum zu dem Artikel hinzugefügt. Der Artikel wird als reserviert markiert.

**Tatsächliches Ergebnis**

Die eingegebenen Informationen werden korrekt ausgelesen und an der richtigen Stelle in die Datenbank eingetragen. Außerdem wird automatisch das aktuelle Datum mit Uhrzeit als Eingangsdatum zu dem Artikel hinzugefügt. Der Artikel wird als reserviert markiert.

**Testfall 2:**

Ein neuer Artikel wird im Lager angelegt. Dafür werden folgende Testwerte in der GUI eingegeben

Artikelnummer: 12345

Stückzahl: 10

Artikelbeschreibung: Ein Artikel im Lager, der nicht reserviert ist

Preis: 50

Preisaufschlag: 50

Regal: 1

Fach: 2

Reserviert

Waren

**Erwartetes Ergebnis:**

Die eingegebenen Informationen werden korrekt ausgelesen und an der richtigen Stelle in die Datenbank eingetragen. Außerdem wird automatisch das aktuelle Datum mit Uhrzeit als Eingangsdatum zu dem Artikel hinzugefügt. Der Artikel wird als nicht reserviert markiert.

**Tatsächliches Ergebnis:**

Die eingegebenen Informationen werden korrekt ausgelesen und an der richtigen Stelle in die Datenbank eingetragen. Außerdem wird automatisch das aktuelle Datum mit Uhrzeit als Eingangsdatum zu dem Artikel hinzugefügt. Der Artikel wird als nicht reserviert markiert.

**Artikelinfo Testfälle**

**Testfall 1:**

Anzeigen des Artikels mit der Artikelnummer 1234

**Erwartetes Ergebnis:**

Die Artikelinformationen des Artikels mit der Artikelnummer 1234 werden angezeigt.

**Tatsächliches Ergebnis:**

Die Artikelinformationen des Artikels mit der Artikelnummer 1234 werden angezeigt.

**Testfall 2:**

Anzeigen des Artikels mit der Artikelnummer 12345AB

**Erwartetes Ergebnis:**

Es werden keine Artikelinformationen angezeigt, da kein Artikel mit dieser Artikelnummer in der Datenbank existiert.

**Tatsächliches Ergebnis:**

Es werden keine Artikelinformationen angezeigt, da kein Artikel mit dieser Artikelnummer in der Datenbank existiert.

**Wareneingang Testfälle**

**Testfall 1:**

Von dem Artikel mit der Artikelnummer 1234 werden 10 Stück ins Lager aufgenommen. Die Werte werden in der GUI eingetragen.

**Erwartetes Ergebnis:**

Die Stückzahl auf der Datenbank erhöht sich um 10.

**Tatsächliches Ergebnis:**

Stückzahl der Datenbank erhöht sich um 10.

**Testfall 2:**

Von dem Artikel mit der Artikelnummer 1234 werden 10A Artikel ins Lager aufgenommen. Die Werte werden in der GUI eingetragen.

**Erwartetes Ergebnis:**

Fehlermeldung und Abbruch des Vorgangs, da keine korrekte Zahl als Stückzahl eingegeben wurde.

**Tatsächliches Ergebnis:**

Fehlermeldung und Abbruch des Vorgangs, da keine korrekte Zahl als Stückzahl eingegeben wurde.

**Warenausgang Testfälle**

**Testfall 1:**

Von dem Artikel mit der Artikelnummer 1234 werden 10 Stück aus dem Lager ausgebucht. Die Werte werden durch die GUI eingetragen.

**Erwartetes Ergebnis:**

Der Lagerbestand des Artikels mit der Artikelnummer 1234 wird um 10 Stück reduziert.

**Tatsächliches Ergebnis:**

Der Lagerbestand des Artikels mit der Artikelnummer 1234 wird um 10 Stück reduziert.

**Testfall 2:**

Von dem Artikel mit der Artikelnummer 1234 werden 1000 Stück aus dem Lager ausgebucht. Die Werte werden durch die GUI eingetragen.

**Erwartetes Ergebnis:**

Der Lagerbestand des Artikels mit der Artikelnummer 1234 wird nicht reduziert und es wird eine Fehlermeldung ausgegeben, da nicht so viele Artikel aus dem Lager ausgebucht werden können. Der Lagerbestand des Artikels ist nicht hoch genug und es können nicht mehr Artikel ausgebucht werden, als sich tatsächlich im Lager befinden.

**Tatsächliches Ergebnis:**

Der Lagerbestand des Artikels mit der Artikelnummer 1234 wird nicht reduziert und es wird eine Fehlermeldung ausgegeben.

# Übergabe

Eine ausführbare Datei der Anwendung, sowie die Dateien mit dem Quellcode wurden dem Kunden auf einem USB-Stick übergeben. Dazu wurden bei dem Kunden eine Präsentation der Funktionalitäten und eine Schulung durchgeführt. Die Schulung wurde unter anderem von den Mitarbeitern im Lager besucht, so dass eventuelle Fragen von den Benutzern schon vor der Benutzung und Einführung beim Kunden geklärt werden konnten. Anhand der Kundendokumentation kann der Kunde die Funktionen und die präsentierten Inhalte nachlesen.

# Fazit

## Soll- ist Vergleich

Bei einer rückblickenden Betrachtung des IHK-Abschlussprojektes, kann festgehalten werden, dass alle zuvor festgelegten Anforderungen erfüllt wurden. Zu Beginn des Projekts wurde im Abschnitt Projektphasen ein Projektplan erstellt, welcher eingehalten wurde. In der folgenden Tabelle werden die geplante Zeit und die tatsächlich benötigte Zeit gegenübergestellt. An drei Stellen wurde geringfügig von der geplanten Zeit abgewichen, dies hatte jedoch kein Einfluss auf die Gesamtzeit, so dass das Projekt in 80 Stunden umgesetzt werden konnte. Der Entwurf des Projekts hat etwas weniger Zeit als erwartet in Anspruch genommen, dafür wurde bei der Implementierung mehr Zeit benötigt. Besonders die Datenbankverbindung und das Abfangen falscher Benutzereingaben hat mehr Zeit als erwartet eingenommen. Der zentrale Aspekt des Projekts ist die Datenbank und es war wichtig, die Zeit darauf zu verwenden, um fehlerhafte Interaktionen und Datenbankabstürze zu verhindern.

|  |  |  |  |
| --- | --- | --- | --- |
| Projektphase | Soll | Ist | Differenz |
| Analysephase | 7 h | 8 h | + 1 h |
| Entwurfsphase | 17 h | 14 h | - 3 h |
| Implementierungsphase | 28 h | 30 h | + 2 h |
| Abnahme und Einführung | 1 h | 1h | 0 h |
| Erstellen der Dokumentation | 27 h | 27 h | 0 h |
| Gesamt | 80 h | 80 h | 0 h |

## Gelerntes

Im Zuge des Projektes konnte der Autor wertvolle Erfahrung über die Planung und Umsetzung eines Software Projektes sammeln. Der Autor lernte wie man ein Projekt von Grund auf plant und worauf dabei wertgelegt werden sollte.

Darüber hinaus war es für den Autor eine gute Übung ein Projekt in einer festgelegten Zeit fertigzustellen. Der Aspekt der Zeitbegrenzung erforderte ein hohes Maß an Organisation und Überblick über die erledigten und ausstehenden Aufgaben. Der Autor ist sich sicher, das Gelernte aus diesem Projekt nahtlos in das nächste Projekt einbeziehen zu können und sicher, dass er persönlich von dem Projekt profitiert hat. Die vielen kleinen Hürden und Probleme, die schnell behoben werden mussten, haben gezeigt, wie wichtig es ist, von Anfang an einen guten Plan und effizientes Projektmanagement zu haben. Weitere Schlüsselerkenntnisse sind die hohen Stellenwerte von Kommunikation mit dem Auftraggeber und Dokumentation, damit der Planungs- und Entwicklungsprozess transparent und verständlich dargelegt ist .

Das Arbeiten mit WPF hat sich als sehr angenehm herausgestellt und das Framework hat den gesamten Arbeitsprozess stark erleichtert. Besonders das Erstellen von Benutzeroberflächen konnte in einem sehr einfachen Prozess erreicht werden.

## Ausblick

Obwohl alle Anforderungen und Funktionalitäten umgesetzt wurden kann das Programm noch vielfältig erweitert werden. Sobald die Software im Lager eingesetzt wird könnten nach Feedback mit den Lagerarbeitern, welche das Programm täglich benutzen, noch mehr Anforderungen oder Wünsche entstehen. Es bedarf einer Menge Rückmeldung durch Anwender, um eine effiziente Benutzeroberfläche designe zu können. Dies war im Rahmen des Projektes nur bedingt gegeben. Das Projekt als solches bietet eine gute Basis und eine Vielzahl an Funktionen, welche durch die IT-Abteilung des Kunden leicht modifiziert und erweitert werden können.

Es ist wichtig, die Anwendung weiter unter Realbedingungen zu testen und prüfen damit auch in Zukunft Fehler und mögliche Fehlerquellen schnell identifiziert und angepasst werden können. Die Sicherheit und Verfügbarkeit von Systemen spielen in eine große Rolle. Der Kunde wird die Anwendung erweitern und die Möglichkeiten zur Bearbeitung von Datensätzen in der GUI umsetzten.

# Anhang

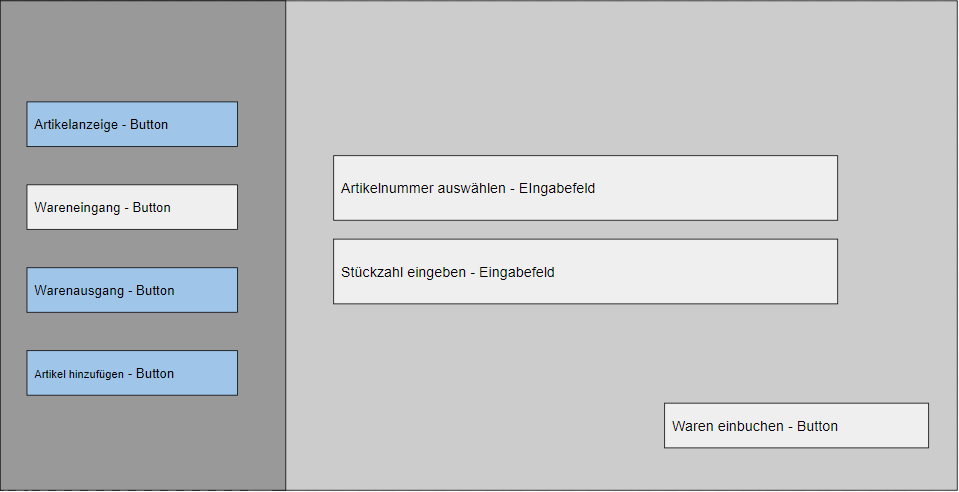


Abbildung 1 - Skizze des Wareneingangs

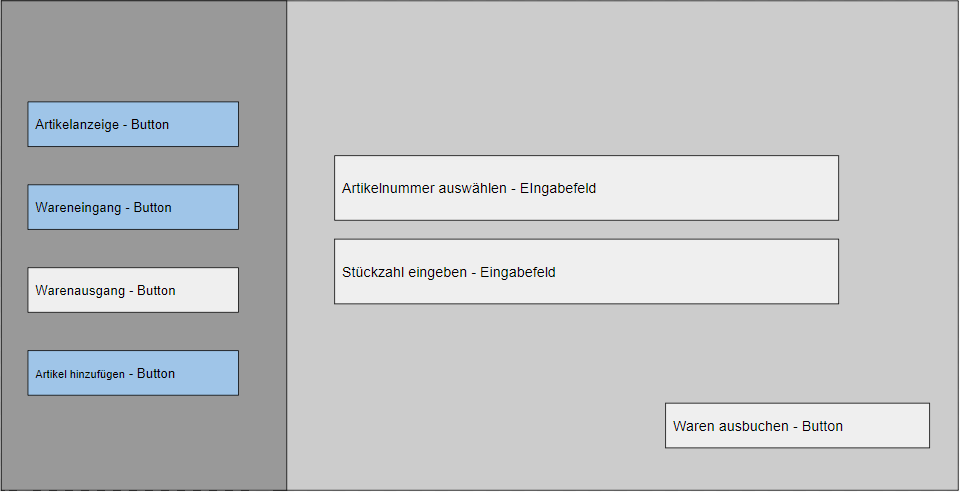


Abbildung 2 - Skizze des Warenausgangs

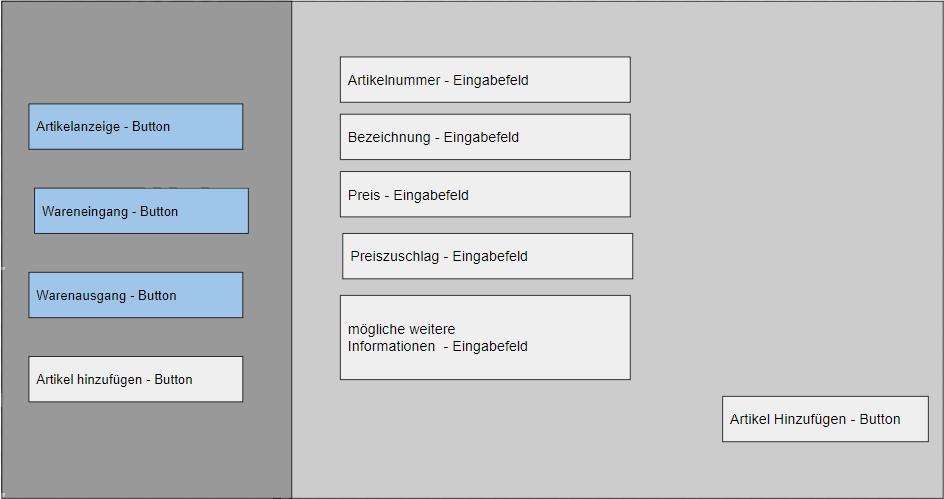


Abbildung 3 - Skizze Artikel Hinzufügen

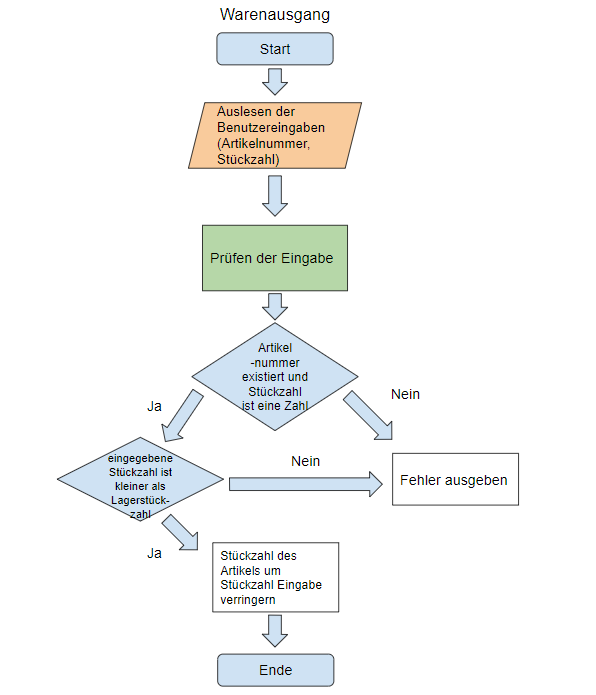


Abbildung 4 Warenausgang PAP

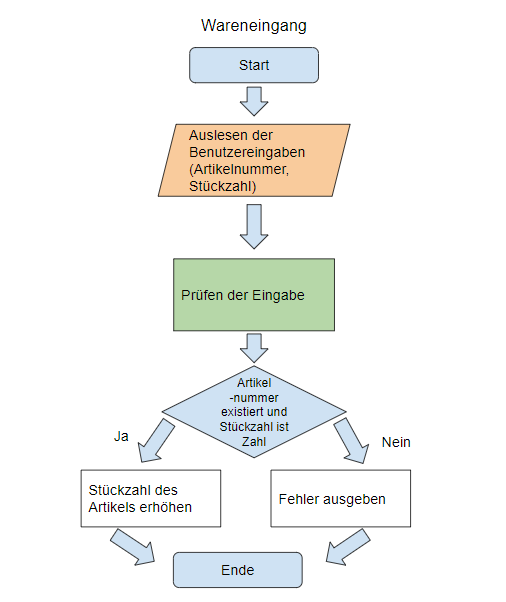


Abbildung 5 Wareneingang PAP

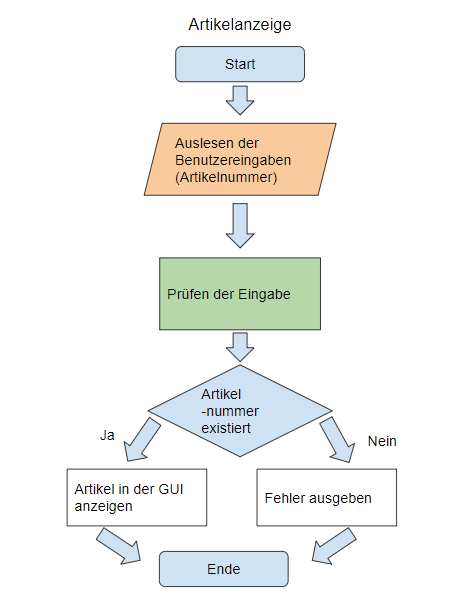


Abbildung 6 Artikelanzeige PAP

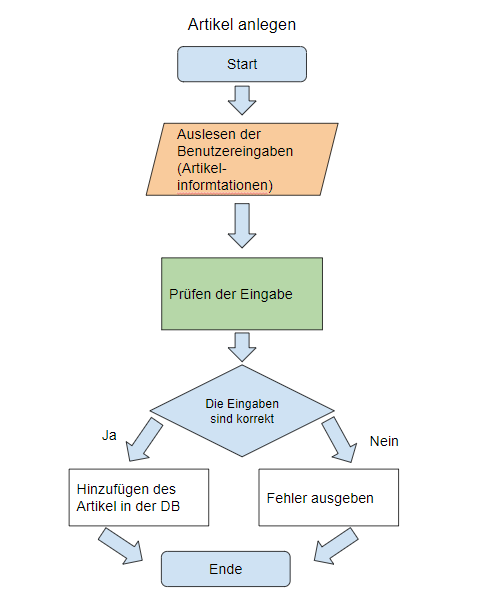


Abbildung 7 Artikel anlegen PAP

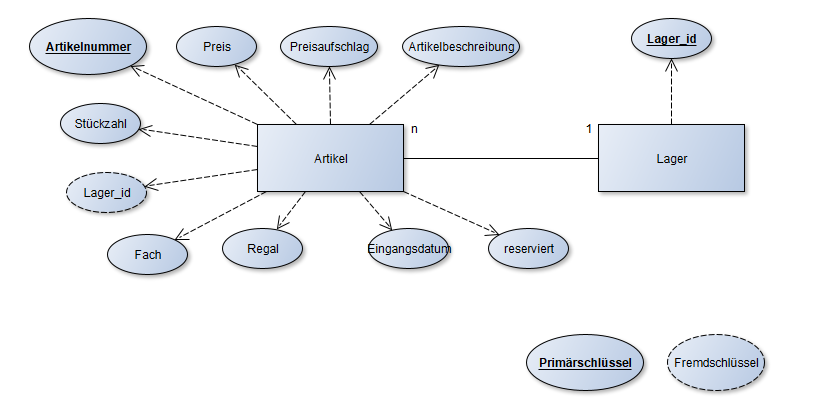


Abbildung 8 Datenbankdesign ERM

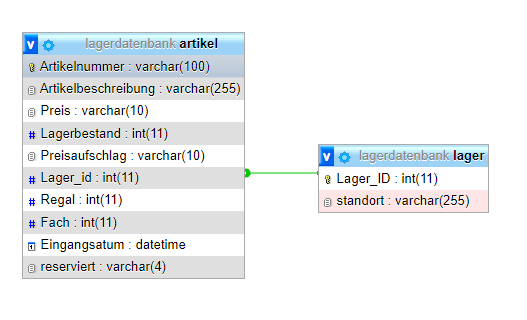


Abbildung 9 Datenbankdesign

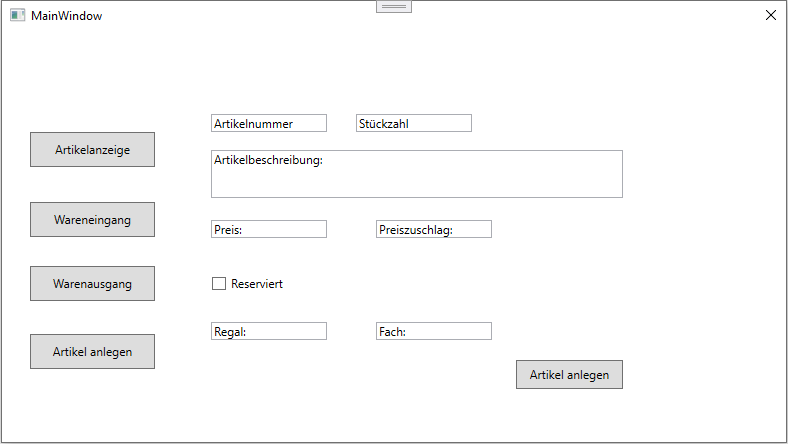


Abbildung 10 GUI Artikel anlegen

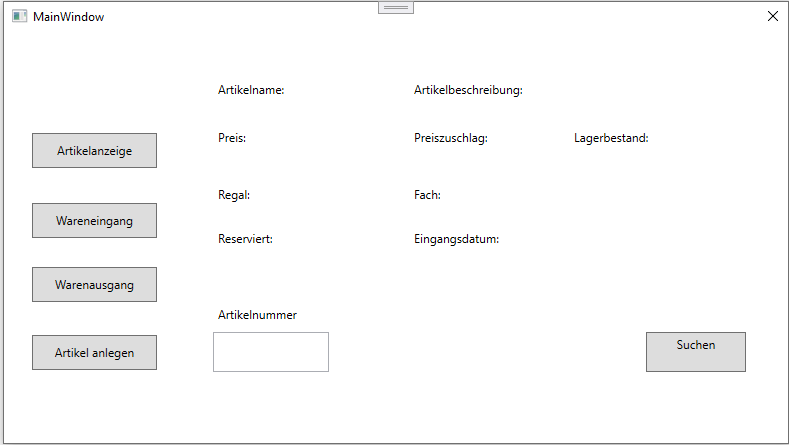


Abbildung 11 GUI Artikel anzeigen

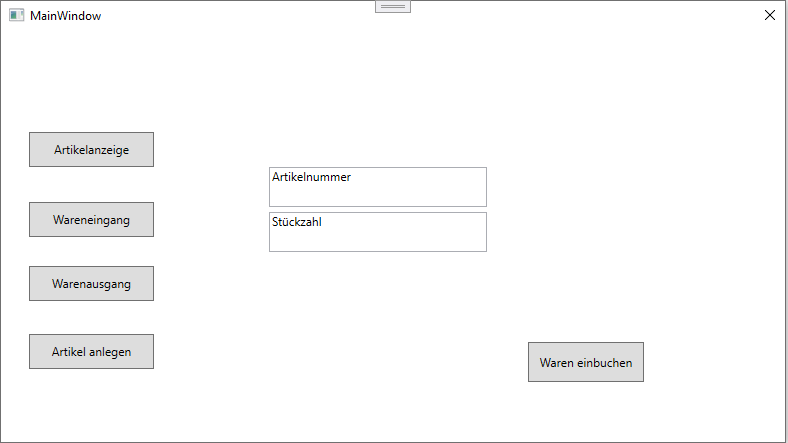


Abbildung 13 GUI Wareneingang

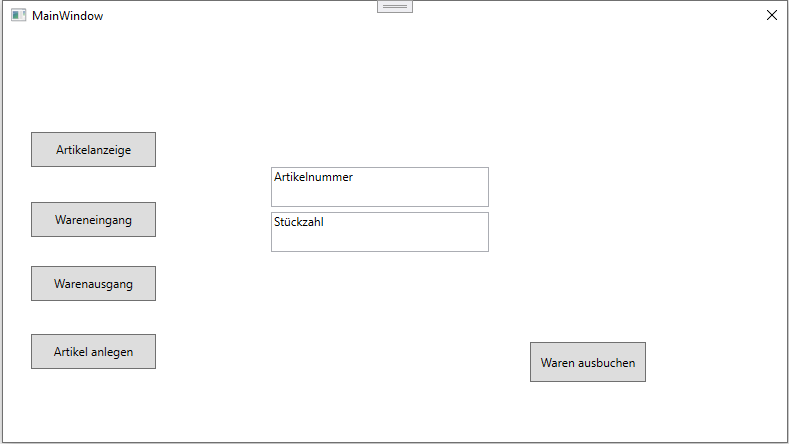


Abbildung 14 GUI Warenausgang

# Quellen

## Tools

* Versionskontrolle
  + <https://github.com/>
* Erstellen der Diagramme
  + <https://docs.google.com/>
* Erstellen der ER-Modelle
  + <https://www.yworks.com/products/yed>

## Code-bedingte Quellen

* Recherche Site WPF
  + <https://learn.microsoft.com/de-de/dotnet/desktop/wpf/app-development/navigation-overview?view=netframeworkdesktop-4.8>
* Recherche Frame WPF
  + <https://learn.microsoft.com/de-de/dotnet/desktop/wpf/windows/?view=netdesktop-7.0>
* SQL Benutzer
  + <https://portal.d-velop.de/documentation/installupdatemanuald3/latest/de/haeufig-gestellte-fragen/wie-erstelle-ich-einen-benutzer-in-microsoft-sql-server>
* SQL Update Nachlesen
  + <https://www.w3schools.com/sql/sql_update.asp>
  + <http://sql.lernenhoch2.de/lernen/sql-anfanger/update-daten-aktualisieren/>
* SQL Insert Into Nachlesen
  + <https://www.1keydata.com/de/sql/sql-insert-into.php>
* SQL Select Nachlesen
  + <https://www.w3schools.com/sql/sql_ref_select.asp>
* .Net Framework
  + <https://www.giga.de/downloads/windows-10/specials/was-ist-net-framework-einfach-erklaert/#:~:text=Das%20.,genannt)%2C%20Klassenbibliothek%20und%20Compiler>.

## Doku-bedingte Quellen

* Warenwirtschaftssystem Nachschlagen
  + <https://www.erp-system.de/warenwirtschaftssystem/>
* ERM Nachschlagen
  + <https://www.datenbanken-verstehen.de/datenmodellierung/entity-relationship-modell/>
* Primärschlüssel Nachlesen
  + <https://www.datenbanken-verstehen.de/datenmodellierung/primaerschluessel/#:~:text=Der%20Prim%C3%A4rschl%C3%BCssel%20kommt%20in%20relationalen,besitzen%20alle%20Tabellen%20einen%20Prim%C3%A4rschl%C3%BCssel>.
* Fremdschlüssel Nachlesen
  + <https://www.ibm.com/docs/de/db2/10.5?topic=schemas-primary-foreign-keys>
* Programmablaufplan Nachlesen
  + <https://project-base.org/flussdiagramm/programmablaufplan/>
* Dynamische Lagerhaltung Definition
  + <https://www.bito.com/de-de/fachwissen/artikel/was-ist-dynamische-lagerhaltung/>
* Getter und Setter Beschreibung
  + <https://studyflix.de/informatik/java-methoden-1901>
* Drag and Drop Beschreibung
  + <https://www.e-teaching.org/materialien/glossar/drag-drop>
* Beschreibung Software Glossar
  + <https://it-service.network/it-lexikon/software>

# Kundendokumentation

**Installation**

Damit die Anwendung auf ihre Windows Computer funktioniert muss das .NET Framework installiert sein. Des Weiteren wird ein Server mit XAMPP und mysql benötigt, welcher sich im selben Netzwerk befindet. Die Anwendung erhalten Sie als .exe auf einem USB-Stick

Starten sie XAMPP und öffnen sie die Benutzeroberfläche von phpMyAdmin. Erstellen sie eine neue Datenbank mit dem Name Lagerdatenbank. Nachdem sie die Datenbank erstellt und ausgewählt haben gehen sie zum Menüpunkt SQL. Hier können eigene SQL Befehle eingegeben werden. Geben sie die folgenden grau hinterlegten Befehle ein.

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" \t "mysql_doc) [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" \t "mysql_doc)(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)

);

Damit von der Anwendung auf die Datenbank zugegriffen werden kann wird ein Benutzer mit entsprechenden Rechten benötigt. Um einen neuen Benutzer zu erstellen geben sie den folgenden Befehl ein. Dieser Befehl erstellt einen neuen lokalen Benutzer mit dem Namen Lagerarbeiter.

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

Damit der Benutzer über die Rechte verfügt mit der Datenbank zu interagieren wird ein weiterer Befehl benötigt. Der folgende Befehl gibt dem Benutzer Lagerarbeiter vollen Zugriff auf die Datenbank.

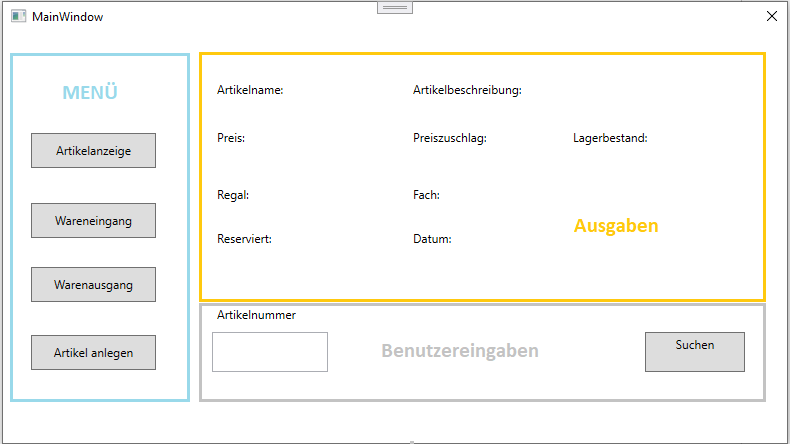
GRANT ALL PRIVILEGES ON lagerdatenbank . \* TO 'Lagerarbeiter'@'localhost';

pGBenutz

**Artikelanzeige**

Beim Start des Programms öffnet sich standartmäßig das Fenster zum Anzeigen von Artikelinformationen. Dieses Fenster ermöglicht das Einsehen von Artikeln aus der Datenbank. Links im Bildrand befindet sich Ein Menü, welches das Navigieren zwischen den Funktionalitäten ermöglicht.

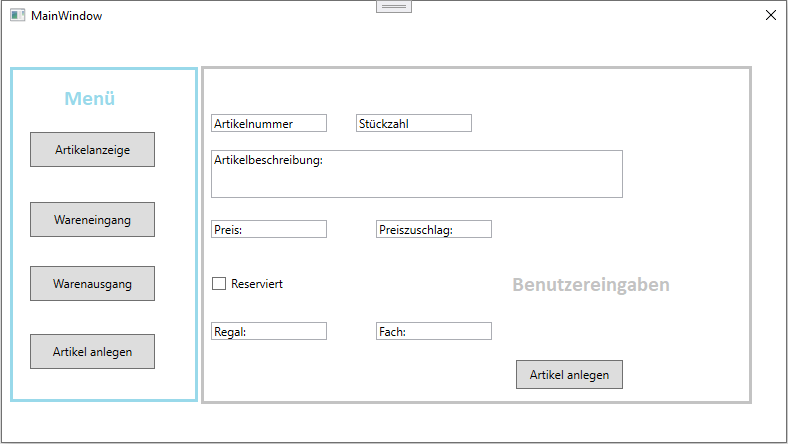
Die Felder in dem orangenen Rahmen sind die Ausgaben. Mit den Benutzereingaben in dem grauen Rahmen kann mit dem Programm interagiert werden. Um einen Artikel aus dem Lager anzuzeigen muss eine Artikelnummer in das Eingabefeld eingetragen werden und mit einem Klick auf den ‚Suchen‘ Knopf wird der Vorgang gestartet. Danach werden die Artikelinformationen in den Ausgabefeldern angezeigt.



1. Abbildung 1 Artikelanzeige

**Anlegen eines neuen Artikels**

Ein neuer Artikel kann auf dieser Seite auf der Datenbank angelegt werden.

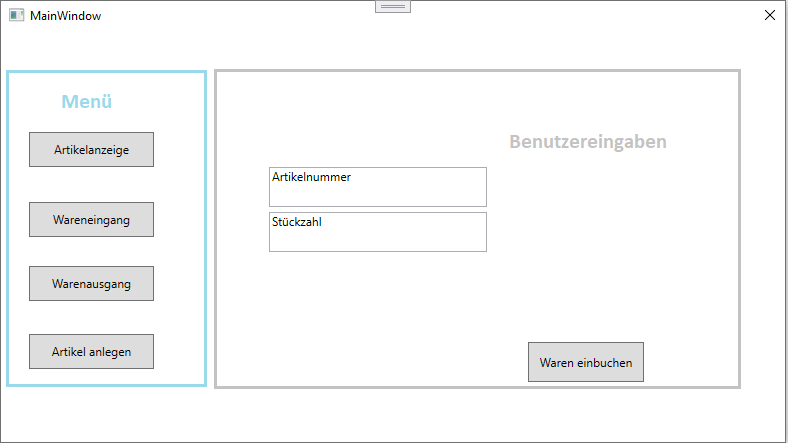


1. Abbildung 4 Artikel hinzufügen

Um einen Artikel anzulegen müssen in die Textfelder die Artikelinformationen eingetragen werden. Die Beschriftungen der Textfelder geben an, was dort eingetragen werden soll. Im Feld Artikelnummer wird die Artikel- oder Seriennummer eingetragen. Die Artikelnummer kann aus Zahlen und Buchstaben bestehen und darf nicht länger als 100 Zeichen sein. Im Feld Artikelbeschreibung kann eine kurze Beschreibung des Artikels eingetragen werden. Dies ist nicht zwingend notwendig Der Text kann nicht länger als 255 Zeichen sein. Ist das Häkchen bei dem Feld ‚Reserviert‘ aktiv wird der eingetragene Artikel in der Datenbank als Reserviert markiert. Standardmäßig wird ein Artikel als nicht reserviert gespeichert. Im Feld Preis und Preiszuschlag werden jeweils Beträge in Euro eingetragen. Sind alle Informationen eingetragen kann der Vorgang mit dem Knopf ‚Artikel anlegen‘ ausgeführt werden.

**Wareneingang**

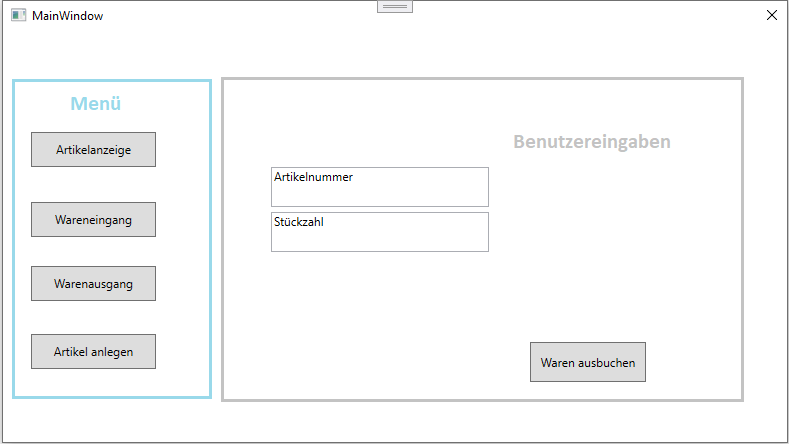
Wird im Menü auf den Knopf Wareneingang geklickt öffnet sich dieses Fenster. Hier können eine Artikelnummer und eine Stückzahl eingetragen werden. In das Eingabefeld ‚Stückzahl‘ dürfen nur Zahlen eingetragen werden. Bei Falscheingabe wird im linken unteren Bildrand eine Fehlermeldung ausgegeben.



1. Abbildung 5Wareneingang

**Warenausgang**

Wird im Menü auf den Knopf Warenausgang geklickt öffnet sich dieses Fenster. Die Seite ist vom Aufbau identisch mit der Wareneingangsseite. Es können eine Artikelnummer und eine Stückzahl eingetragen werden. In Das Eingabefeld ‚Stückzahl‘ dürfen nur Zahlen eingetragen werden.

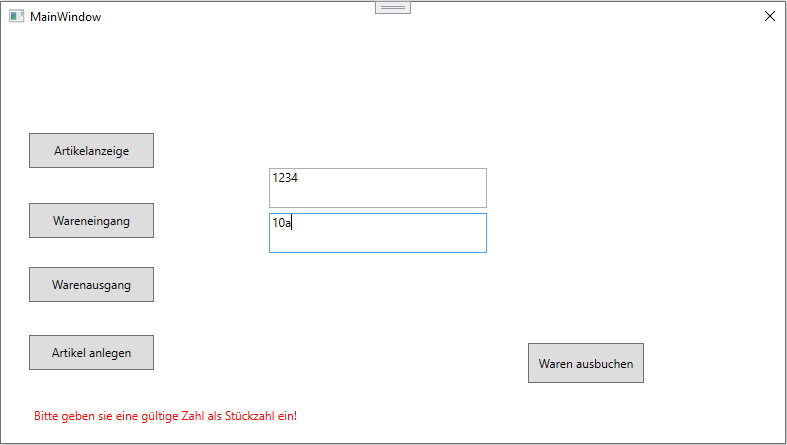


1. Abbildung 6 Warenausgang

**Fehlermeldungen**

Im linken unteren Bildschirm werden Fehlermeldungen ausgegeben. Es kann aus unterschiedlichen Gründen zu Fehlermeldungen kommen. Mögliche Ursachen für Fehlermeldungen sind keine aktive Verbindung mit der Datenbank, fehlerhafte oder fehlende Eingabe.

Hier wird eine Fehlermeldung ausgegeben, da bei ausbuchen eines Artikels keine gültige Zahl eingeben wurde.



# 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. |
| Windows Forms | GUI-Toolkit im .Net-Framework |
| 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 | Engl Site. Ein Programmbaustein in WPF zur Anzeige von Inhalten mit weniger Funktionen als ein Fenster |
| 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 |
| Server | Eine Hardware oder eine Software beziehungsweise der Kombination aus beidem, die Ressourcen, Dienste oder Daten für andere Rechner oder Programme bereitstellt |
| Insert into | Befehl in SQL, welcher einen neuen Eintrag in einer Tabelle erzeugt |
| Int | Datentyp zum speicher einer Ganzzahl |
| Boolean | Datentyp, welcher nur wahr(true) oder fasch(false) sein kann |
| PAP | Programmablaufplan, wird auch Flussdigagramm genannt |
| Github | Programm zur Versionskontrolle |
| MYSQL.DATA | Bibliothek für Datenbankverbindungen |
| SaaS | Software as a Service |
| Hardware | mechanische und elektronische Ausstattung eines Systems |
| Software | alle nicht technisch-physikalischen Funktionsbestandteile eines Computers, insbesondere Computerprogramme |
| ERM | Entity-Relationship-Modell |
| Drag and Drop | Drag and Drop ist eine Arbeitstechnik in grafischen Benutzeroberflächen, bei der Objekte direkt mit dem Mauszeiger angefasst, verschoben ("drag") und in einem Zielbereich losgelassen ("drop") werden können. |
| USB | Universal Serial Bus |
| Parameter | Als Parameter bezeichnet man in der Informatik die Platzhalter für die Eingabewerte einer Methode oder Funktion. |
| .exe | Dateiendung einer ausführbaren Datei bei dem Windows Betriebssystem |
| Feedback | Rückmeldung |
| Frame | Deutsch: Rahmen. In WPF ein Element zum Anzeigen von Elementen der Benutzeroberfläche |
| Content | Inhalt, der in den Frame geladen wird |
| Performance | Leistung |
| MySqlCommand | Befehle aus WPF, der auf SQL |

# Quelltextdokumentation

**Artikel.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Warenwirtschaftssystem

{

internal class Artikel

{

//Klassenattribute

private string artikelnummer;

private string artikelbeschreibung;

private string preis;

private int stückzahl;

private string preisaufschlag;

private int lagerId;

private int regal;

private int fach;

private string Datum;

private string reserviert;

public Artikel(string artikelnummer, string artikelbeschreibung, string preis, int stückzahl, string preisaufschlag, int lagerId, int regal, int fach, string datum, string reserviert)

{

this.Artikelnummer = artikelnummer;

this.Artikelbeschreibung = artikelbeschreibung;

this.Preis = preis;

this.Stückzahl = stückzahl;

this.Preisaufschlag = preisaufschlag;

this.LagerId = lagerId;

this.Regal = regal;

this.Fach = fach;

this.Datum1 = datum;

this.Reserviert = reserviert;

}

public Artikel()

{

}

//getter und sett für alle Klassenattribute

public string Artikelnummer { get => artikelnummer; set => artikelnummer = value; }

public string Artikelbeschreibung { get => artikelbeschreibung; set => artikelbeschreibung = value; }

public string Preis { get => preis; set => preis = value; }

public int Stückzahl { get => stückzahl; set => stückzahl = value; }

public string Preisaufschlag { get => preisaufschlag; set => preisaufschlag = value; }

public int LagerId { get => lagerId; set => lagerId = value; }

public int Regal { get => regal; set => regsal = value; }

public int Fach { get => fach; set => fach = value; }

public string Datum1 { get => Datum; set => Datum = value; }

public string Reserviert { get => reserviert; set => reserviert = value; }

}

}

**Artikel\_anlegen.xaml**

<Page x:Class="Warenwirtschaftssystem.Artikel\_anlegen"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Warenwirtschaftssystem"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Artikel\_anlegen">

<Grid>

<Button Content="Artikel anlegen" HorizontalAlignment="Left" Margin="349,309,0,0" VerticalAlignment="Top" Height="29" Width="107" Click="Button\_Click"/>

<TextBox Name="textbox\_Artikelnummer" ToolTip="Artikelnummer" Text="Artikelnummer" HorizontalAlignment="Left" Margin="44,63,0,0" VerticalAlignment="Top" Width="116" MaxLength="15"/>

<TextBox Name="textbox\_Stückzahl" ToolTip="Stückzahl" Text="Stückzahl" HorizontalAlignment="Left" Margin="189,63,0,0" VerticalAlignment="Top" Width="116"/>

<TextBox Name="textbox\_Preisa" Text="Preis: " HorizontalAlignment="Left" Margin="44,169,0,0" VerticalAlignment="Top" Width="116"/>

<TextBox Name="textbox\_Preiszuschlag" Text="Preiszuschlag: " HorizontalAlignment="Left" Margin="209,169,0,0" VerticalAlignment="Top" Width="116"/>

<TextBox x:Name="textbox\_Regal" Text="Regal: " HorizontalAlignment="Left" Margin="44,271,0,0" VerticalAlignment="Top" Width="116" TextChanged="textbox\_Regal\_TextChanged"/>

<TextBox Name="textbox\_lFach" Text="Fach: " HorizontalAlignment="Left" Margin="209,271,0,0" VerticalAlignment="Top" Width="116" InputScope="Number"/>

<TextBox Name="textbox\_Artikelbeschreibung" Text="Artikelbeschreibung:" HorizontalAlignment="Left" Margin="44,99,0,0" VerticalAlignment="Top" Width="412" Height="48" MaxLength="255"/>

<CheckBox Name="checkBoxReserviert" Content="Reserviert" HorizontalAlignment="Left" Margin="44,225,0,0" VerticalAlignment="Top"/>

</Grid>

</Page>

**Artikel\_anlegen.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Warenwirtschaftssystem

{

/// <summary>

/// Interaktionslogik für Artikel\_anlegen.xaml

/// </summary>

public partial class Artikel\_anlegen : Page

{

public Artikel\_anlegen()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

string artikelnummer = textbox\_Artikelnummer.Text;

string preis = textbox\_Preisa.Text;

string artikelbeschreibung = textbox\_Artikelbeschreibung.Text;

string preisaufschlag = textbox\_Preiszuschlag.Text;

string regal = textbox\_Regal.Text;

string fach = textbox\_Regal.Text;

string stückzahl = textbox\_Stückzahl.Text;

string reserviertString = "Nein";

bool istReserviert = (checkBoxReserviert.IsChecked == true);

if (istReserviert)

{

reserviertString = "Ja";

}

string lagerId = "1";

Datenbankverbindung verbindung = new Datenbankverbindung();

verbindung.setConnection();

verbindung.fügeNeuenArtikelZurDatenbank(artikelnummer, artikelbeschreibung, preis, stückzahl, preisaufschlag, lagerId, regal, fach, reserviertString);

verbindung.closeConnection();

}

private void CheckBox\_Checked(object sender, RoutedEventArgs e)

{

}

private void checkBoxReserviert\_Checked(object sender, RoutedEventArgs e)

{

}

private void textbox\_Regal\_TextChanged(object sender, TextChangedEventArgs e)

{

}

}

}

**Artikelanzeige.xaml**

<Page x:Class="Warenwirtschaftssystem.Artikelanzeige"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Warenwirtschaftssystem"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Artikelanzeige">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="5\*"/>

<RowDefinition Height="4\*"/>

</Grid.RowDefinitions>

<Button Content="Suchen&#xD;&#xA;" HorizontalAlignment="Left" Margin="477,74,0,0" VerticalAlignment="Top" Click="Button\_Click" Height="40" Width="100" Grid.Row="1"/>

<TextBox HorizontalAlignment="Left" Margin="44,74,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="116" Height="40" TextChanged="TextBox\_TextChanged" Name="eingabe\_Artikelnummer" Grid.Row="1"/>

<Label Content="Artikelnummer" HorizontalAlignment="Left" Margin="44,43,0,0" Grid.Row="1" VerticalAlignment="Top" Width="116"/>

<Label Name="labelArtikelname" Content="Artikelname: " HorizontalAlignment="Left" Margin="44,24,0,0" VerticalAlignment="Top" Width="116"/>

<Label Name="labelPreis" Content="Preis: " HorizontalAlignment="Left" Margin="44,72,0,0" VerticalAlignment="Top" Width="116"/>

<Label Name="labelPreiszuschlag" Content="Preiszuschlag: " HorizontalAlignment="Left" Margin="240,72,0,0" VerticalAlignment="Top" Width="116"/>

<Label Name="labelRegal" Content="Regal: " HorizontalAlignment="Left" Margin="44,129,0,0" VerticalAlignment="Top" Width="116"/>

<Label Name="labelFach" Content="Fach: " HorizontalAlignment="Left" Margin="240,129,0,0" VerticalAlignment="Top" Width="116"/>

<Label Name="labelArtikelbeschreibung" Content="Artikelbeschreibung:" HorizontalAlignment="Left" Margin="240,24,0,0" VerticalAlignment="Top" Width="412" Height="48"/>

<Label Name="labelReserviert" Content="Reserviert: " HorizontalAlignment="Left" Margin="44,173,0,0" VerticalAlignment="Top" Width="116"/>

<Label Name="labelDatum" Content="Eingangsdatum: " HorizontalAlignment="Left" Margin="240,173,0,0" VerticalAlignment="Top" Width="160"/>

<Label Name="labelLagerbestand" Content="Lagerbestand: " HorizontalAlignment="Left" Margin="400,72,0,0" VerticalAlignment="Top" Width="116"/>

</Grid>

</Page>

**Artikelanzeige.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Warenwirtschaftssystem

{

public partial class Artikelanzeige : Page

{

public Artikelanzeige()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

//artikelnummer wird initialisiert und mit Benutzereingabe deklarasiert

string artikelnummer = eingabe\_Artikelnummer.Text;

//Datenbankverbindung wird aufgebaut

Datenbankverbindung connection = new Datenbankverbindung();

//Klasse Feedback wird initialisiert

Feedback feedback = new Feedback();

//die Funktion gibt bei erfolgreichem Aufbau der Datenbankverbindung true zürück

if (connection.setConnection())

{

//Objekt der Klasse Artikel wird angelegt und mit Werten aus der Datenbank befüllt

Artikel artikel = connection.gebeArtikelFürArtikelnummerZurück(artikelnummer);

//es wurde ein Artikel mit der eingegeben Artikelnummer gefunden und zurückgegeben

if (artikel != null)

{

labelArtikelname.Content = "Artikelnummer: " + artikel.Artikelnummer;

labelArtikelbeschreibung.Content = "Artikelbeschreibung: " + artikel.Artikelbeschreibung;

labelPreis.Content = "Artikelpreis: " + artikel.Preis;

labelPreiszuschlag.Content = "Preisaufschlag: " + artikel.Preisaufschlag;

labelFach.Content = "Fach: " + artikel.Fach;

labelRegal.Content = "Regal: " + artikel.Regal;

labelReserviert.Content = "Reserviert: "+ artikel.Reserviert;

labelDatum.Content = "Eingangsdatum: " + artikel.Datum1;

labelLagerbestand.Content = "Lagerbestand: " + artikel.Stückzahl;

feedback.erstelleErfolgsmeldung("Ausführen Erfolgreich!");

}

//es wurde kein Artikel mit der eingegeben Artikelnummer gefunden

else

{

feedback.erstelleFehlermeldung("Kein Artikel mit dieser Artikelnummer gefunden");

}

//Datenbankverbindung wird geschlossen

connection.closeConnection();

}

//es konnte keine Verbindung zur Datenbank aufgestellt werden

else

{

feedback.erstelleFehlermeldung("Es konnte keine Datenbankverbindung aufgebaut werden!");

}

}

private void TextBox\_TextChanged(object sender, TextChangedEventArgs e)

{

}

}

}

**Datenbankverbindung.cs**

using MySql.Data.MySqlClient;

using Org.BouncyCastle.Utilities.Collections;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Printing;

using System.Text;

using System.Threading.Tasks;

namespace Warenwirtschaftssystem

{

class Datenbankverbindung

{

//Verbindung zur SQL Datenbank in einer Variable

private MySqlConnection connection;

//Verbindungsstring, der alle Informationen erhält, um die Datenbankverbindung herzustellen

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

//getter, der die Verbindung zurückgibt

public MySqlConnection getConnection()

{

//gibt die Datenbankverbindung zurück

return this.connection;

}

//Mehtode, welche versucht eine Datenbankverbindung aufzubauen

public bool setConnection()

{

try

{

//erstellt die Datenbankverbindung mit den Informationen aus dem connectionString

this.connection = new MySqlConnection(connectionString);

//öffnet die Verbindung

this.connection.Open();

//gibt wahr zurück, wenn die Verbindung aufgebaut wurde

return true;

}

catch (Exception e)

{

}

//gibt falls zurück, falls keine Datenbankverbindung aufgebaut werden konnte

return false;

}

//gibt den Status der Datenbankverbindung zurück

public bool checkConnection()

{

//bei Rückgabewert 'true' besteht eine aktive Verbindung zur Datenbank

if (this.connection != null) { return true; }

//bei Rückgabewert 'false' besteht keine aktive Verbindung zur Datenbank

return false;

}

//Methode, welche für die übergebene Artikelnummer die Artikelinformationen als Objekt der Klasse Artikel zurückgibt

public Artikel gebeArtikelFürArtikelnummerZurück(string input)

{

//eine neues Objekt der Klasse Artikel wird erstellt

Artikel artikel = new Artikel();

//wenn keine Datenbankverbindung besteht wird ein leerer Artikel zurück gesendet

if (this.checkConnection() == false){ return artikel; }

//SQL Befehl, der den Lagerbestand eines Artikels anzeigt

//Die Lagerbestand des Artikels muss größer 0 sein

string commandString = "SELECT \* FROM artikel " +

"WHERE artikelnummer LIKE "+input+ " AND Lagerbestand > 0";

//der string befel wird in einen SQL Command umgewandelt

MySqlCommand command = new MySqlCommand(commandString, this.connection);

//der SQl Befehl wird ausgeführt und die Ergebnisse werden in dem Object reader gespeichert

MySqlDataReader reader = command.ExecuteReader();

//While Schleife, die durch die Informationen im reader iteriert

while (reader.Read() && reader.HasRows)

{

//Die Artikelinformationen werden aus der Datenbank ausgelesen

//und in einem Objekt der Klasse Artikel gespeichert

artikel.Artikelnummer = reader.GetString(0);

artikel.Artikelbeschreibung = reader.GetString(1);

artikel.Preis = reader.GetString(2);

artikel.Stückzahl = reader.GetInt32(3);

artikel.Preisaufschlag = reader.GetString(4);

artikel.LagerId = reader.GetInt16(5);

artikel.Regal = reader.GetInt32(6);

artikel.Fach = reader.GetInt32(7);

artikel.Datum1 = reader.GetString(8);

artikel.Reserviert = reader.GetString(9);

}

//der Daten

reader.Close();

//das Objekt artikel mit den Artikelinformationen wird zurückgegeben

return artikel;

}

//Methode, welche für die übergeben Artikelnummer die Stückzahl des Artikels anzeigt

public string getLagerbestandVonDatenbankFürArtikelnummer(string artikelNummer)

{

string output = "";

//wenn keine Verbindung mit der Datenbank besteht

if (this.checkConnection() == false) {

output = "";

}

else

{

//wenn die eingegben Artikelnummer nicht der String 'Artikelnummer'

if (!artikelNummer.Equals("Artikelnummer"))

{

//SQL Befehl, der sich für eine bestimmte Artikelnummer

//den Lagerbestand von der Datenbank holt

string commandString = "SELECT Lagerbestand FROM artikel" +

" Where Artikelnummer LIKE " + artikelNummer + "";

//SQl string Befehl wird umgewandelt in ein MySQLCommand

MySqlCommand command = new MySqlCommand(commandString, this.connection);

//der SQL Befehl wird ausgeführt und die Informationen werden

MySqlDataReader reader = command.ExecuteReader();

//solange sich weitere Informationen in dem Objekt reader befinden

while (reader.Read() && reader.HasRows)

{

//die Stückzahl wird zu dem string output addiert

output += reader.GetValue(0);

}

//der reader wird geschlossen

reader.Close();

}

}

//der Ergebniss String wird zurückgegeben

return output;

}

//Methode, welche für die übergebene Artikelnummer und Stückzahl die Stückzahl des Artikels in der Datenbank ändert

public bool setztNeuenLagerbestandFürArtikelnummer(string artikelnummer, int stückzahl)

{

//String, SQL Befehl, welcher den Lagerbestand in der Datenbank

//für eine bestimmte Artikelnummer aktualisiert

string commandString = "UPDATE artikel SET Lagerbestand = " + stückzahl

+ " Where Artikelnummer LIKE " + artikelnummer;

MySqlCommand command = new MySqlCommand(commandString, this.connection);

//der SQL Befehl wird ausgeführt

MySqlDataReader reader = command.ExecuteReader();

reader.Close();

return false;

}

//Methode, welche ein neuen Artikel mit den übergebenen Informationen erstellt

public bool fügeNeuenArtikelZurDatenbank(string artikelNummer, string artikelbeschreibung, string preis, string stückzahl, string preisaufschlag, string lager\_id, string regal, string fach, string reserviert)

{

//SQl Befehl, welcher einen neuen Eintrag in der Tabelle Artikel erstellt.

string commandString = "INSERT INTO artikel(Artikelnummer, Artikelbeschreibung," +

"Lagerbestand, preis, Preisaufschlag, Lager\_id, regal, fach, Eingangsatum," +

" reserviert)" +

" VALUES( "+artikelNummer+ ", '"+ artikelbeschreibung + "', " + stückzahl +

", " + preis + ", " + preisaufschlag + ", " + lager\_id + ", " + regal +

", " + fach + ", CURRENT\_TIMESTAMP" + ", '" + reserviert+"')";

MySqlCommand command = new MySqlCommand(commandString, this.connection);

try

{

//der reader führt den Befehl aus

MySqlDataReader reader = command.ExecuteReader();

//der Befehl wurde erfolgreich durchgeführt und true wird zurückgegeben

return true;

}

catch (Exception e)

{

}

//der Vorgang lief nicht erfolgreich ab und false wird zurückgegeben

return false;

}

//Methode, welche die Verbindug zur Datenbank beendet

public void closeConnection()

{

//Die Datenbankverbindung wird geschlossen

connection.Close();

}

}

}

**Feedback.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Media;

using System.Windows;

namespace Warenwirtschaftssystem

{

internal class Feedback

{

public Feedback()

{

}

private void erstelleRückmeldung(String text, SolidColorBrush brush)

{

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

Window window = Application.Current.MainWindow;

//setzte die Schriftfarbe des labels auf die Farbe

//welcher als Parameter übergeben wurde

(window as MainWindow).label\_Fehlermeldung.Foreground = brush;

//Der Text des Fehlermeldelabes wird gändert

(window as MainWindow).label\_Fehlermeldung.Content = text;

}

public void erstelleFehlermeldung(String text)

{

erstelleRückmeldung(text, new SolidColorBrush(Colors.Red));

}

public void erstelleErfolgsmeldung(String text)

{

erstelleRückmeldung(text, new SolidColorBrush(Colors.Green));

}

}

}

**MainWindow.xaml**

<Window x:Class="Warenwirtschaftssystem.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Warenwirtschaftssystem"

mc:Ignorable="d"

ResizeMode="NoResize"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Button Name="button\_Artikelanlegen" Content="Artikel anlegen" HorizontalAlignment="Left" VerticalAlignment="Top" Height="35" Width="125" Click="legeNeuenArtikelAn" Margin="28,303,0,0"/>

<Button Name="button\_Warenausgang" Content="Warenausgang" HorizontalAlignment="Left" VerticalAlignment="Top" Height="35" Width="125" Click="geheZuWarenausgang" Margin="28,235,0,0"/>

<Button Name="button\_Wareneingang" Content="Wareneingang" HorizontalAlignment="Left" VerticalAlignment="Top" Height="35" Width="125" Click="geheZuWareneingang" Margin="28,171,0,0"/>

<Button Name="button\_Artikelanzeigen" Content="Artikelanzeige" HorizontalAlignment="Left" VerticalAlignment="Top" Height="35" Width="125" Click="zeigeWarenAn" Margin="28,101,0,0"/>

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

<Label Content="" HorizontalAlignment="Left" Margin="72,43,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.056,-0.165" Name="label\_meldung"/>

<Label Name="label\_Fehlermeldung" Foreground="Red" Content="" HorizontalAlignment="Left" Margin="28,370,0,0" VerticalAlignment="Top" Width="747" x:FieldModifier="public"/>

</Grid>

</Window>

**Mainwindow.xaml.cs**

using MySql.Data.MySqlClient;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Warenwirtschaftssystem

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

frame\_Content.Content = new Artikelanzeige();

//button\_Artikelanzeigen.Background = Brushes.Gray;

}

//Läd eine Unterseite in den Content Frame der Hauptseite

private void geheZuWareneingang(object sender, RoutedEventArgs e)

{

//Weiterleitung zur Seite für das Aufnehmen von Waren

frame\_Content.Content = new Wareneingang();

label\_Fehlermeldung.Content = "";

}

private void geheZuWarenausgang(object sender, RoutedEventArgs e)

{

//Weiterleitung zur Seite für das managen von Waren, die das Lager verlassen

frame\_Content.Content = new Warenausgang();

label\_Fehlermeldung.Content = "";

}

private void zeigeWarenAn(object sender, RoutedEventArgs e)

{

//Weiterleitung zur Seite für das Einsehen der Artikel im Lager

frame\_Content.Content = new Artikelanzeige();

label\_Fehlermeldung.Content = "";

}

private void legeNeuenArtikelAn(object sender, RoutedEventArgs e)

{

//Weiterleitung zur Seite für das Anlegen von neuen Artikeln

frame\_Content.Content = new Artikel\_anlegen();

label\_Fehlermeldung.Content = "";

}

private void fehlerMeldung(String text)

{

}

}

}

**Warenausgang.xaml**

<Page x:Class="Warenwirtschaftssystem.Warenausgang"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Warenwirtschaftssystem"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Warenausgang">

<Grid>

<Button Content="Waren ausbuchen" HorizontalAlignment="Left" Margin="362,291,0,0" VerticalAlignment="Top" Width="116" Height="40" Click="Button\_Click"/>

<TextBox Name="textbox\_Artikelnummer" HorizontalAlignment="Left" Margin="103,116,0,0" TextWrapping="Wrap" Text="Artikelnummer" VerticalAlignment="Top" Width="218" Height="40"/>

<TextBox Name="textbox\_Stückzahl" HorizontalAlignment="Left" Margin="103,161,0,0" TextWrapping="Wrap" Text="Stückzahl" VerticalAlignment="Top" Width="218" Height="40"/>

</Grid>

</Page>

**Warenausgang.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Warenwirtschaftssystem

{

/// <summary>

/// Interaktionslogik für Warenausgang.xaml

/// </summary>

public partial class Warenausgang : Page

{

public Warenausgang()

{

InitializeComponent();

}

//Vorgang des Warenausgangs

private void Button\_Click(object sender, RoutedEventArgs e)

{

//Die artikelNummer initalisiert und mit der Benutzereingabe aus der Textbox deklariert

string artikelNummer = textbox\_Artikelnummer.Text;

//Die Anzahl der eingegangen Stücke wird initialisiert und mit der Benutzereingabe aus der Textbox deklariert

string ausgehendeStückzahl\_String = textbox\_Stückzahl.Text;

int aktuelleStückzahl = 0;

bool isNumeric = int.TryParse(ausgehendeStückzahl\_String, out int num);

Feedback feedback = new Feedback();

//die eingegbene Stückzahl ist keine Zahl

if (isNumeric== false)

{

feedback.erstelleFehlermeldung("Bitte geben sie eine gültige Zahl als Stückzahl ein!");

}

else{

//Instanz der Klasse Datenbankverbindung wird erstellt

Datenbankverbindung connection = new Datenbankverbindung();

if (connection.setConnection())

{

//die aktuelle Stückzahl des Artikels wird abgefragt und in einer Variable gespeichert

aktuelleStückzahl = int.Parse(connection.getLagerbestandVonDatenbankFürArtikelnummer(artikelNummer));

//die Anzahl der ausgehenden Artikel ist kleiner oder gleich groß wie die Anzahl der verfügbaren Artikel

if (aktuelleStückzahl - int.Parse(ausgehendeStückzahl\_String) > 0)

{

//abziehen der Stückzahlen

int neueStückzahl = aktuelleStückzahl - int.Parse(ausgehendeStückzahl\_String);

//updaten der Stückzahl in der Datenbank

connection.setztNeuenLagerbestandFürArtikelnummer(artikelNummer, neueStückzahl);

}

//Es wird versucht mehr Waren rauszugeben, als im Lager sind

else

{

//Fehlermeldung anzeigen

feedback.erstelleFehlermeldung("Vorgang abgebrochen. Der Lagerbestand des Artikels mit der Artikelnummer " + artikelNummer + " beträgt nur " + aktuelleStückzahl +"!");

}

}

else

{

feedback.erstelleFehlermeldung("Es konnte keine Datenbankverbindung aufgebaut werden!");

}

}

}

}

}

**Wareneingang.xaml**

<Page x:Class="Warenwirtschaftssystem.Wareneingang"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Warenwirtschaftssystem"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Wareneingang">

<Grid>

<Button Content="Waren einbuchen" HorizontalAlignment="Left" Margin="362,291,0,0" VerticalAlignment="Top" Width="116" Height="40" Click="Button\_Click"/>

<TextBox Name="textbox\_Artikelnummer" HorizontalAlignment="Left" Margin="103,116,0,0" TextWrapping="Wrap" Text="Artikelnummer" VerticalAlignment="Top" Width="218" Height="40"/>

<TextBox Name="textbox\_Stückzahl" HorizontalAlignment="Left" Margin="103,161,0,0" TextWrapping="Wrap" Text="Stückzahl" VerticalAlignment="Top" Width="218" Height="40"/>

</Grid>

</Page>

**Wareneingang.xaml.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Warenwirtschaftssystem

{

/// <summary>

/// Interaktionslogik für Wareneingang.xaml

/// </summary>

public partial class Wareneingang : Page

{

public Wareneingang()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

//Die artikelNummer initalisiert und mit der Benutzereingabe aus der Textbox deklarasiert

string artikelNummer = textbox\_Artikelnummer.Text;

//Die Anzahl der eingegangen Stücke wird initialisiert und mit der Benutzereingabe aus der Textbox deklarasiert

string eingegangeStückzahl\_string = textbox\_Stückzahl.Text;

int aktuelleStückzahl = 0;

int neueStückzahl;

//Quelle https://stackoverflow.com/questions/894263/identify-if-a-string-is-a-number visited: 12.07.23

bool isNumeric = int.TryParse(eingegangeStückzahl\_string, out int num);

//Quelle Ende

//erstelle

Feedback feedback = new Feedback();

//eingegeben Stückzahl ist eine Zahl

if (isNumeric)

{

Datenbankverbindung connection = new Datenbankverbindung();

if (connection.setConnection())

{

string lagerStückzahl\_string = connection.getLagerbestandVonDatenbankFürArtikelnummer(artikelNummer);

if (lagerStückzahl\_string != "")

{

aktuelleStückzahl = int.Parse(lagerStückzahl\_string);

//addieren die hinzugefügten Stücke zu den Stückzahlen

neueStückzahl = aktuelleStückzahl + int.Parse(eingegangeStückzahl\_string);

connection.setztNeuenLagerbestandFürArtikelnummer(artikelNummer, neueStückzahl);

feedback.erstelleErfolgsmeldung("Aktion erfolgreich!");

}

else

{

feedback.erstelleFehlermeldung("Keine gültige Artikelnummer eingegeben");

}

connection.closeConnection();

}

else

{

feedback.erstelleFehlermeldung("Es konnte keine Datenbankverbindung aufgebaut werden!");

}

}

//eingegeben Stückzahl ist KEINE Zahl

else

{

feedback.erstelleFehlermeldung("Bitte geben sie eine Zahl als Stückzahl ein");

}

}

}

}

# Abnahmeprotokoll

**IHK Abschlussprüfung Abnahme-Protokoll Moritz Gramer**

1. Artikel hinzufügen
2. Reservierten Artikel hinzufügen
3. Artikel einbuchen
4. Artikel ausbuchen
5. Artikel anzeigen
6. Reservierten Artikel anzeigen
7. Datenbankverbindung
8. Ungültige Stückzahl Warenausgang
9. Nichtexistierenden Artikel anzeigen
10. Ungültige Artikelnummer Wareneingang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Datum |  | Auszubildender |  | Ausbilder |