**BAVS**

**Buchausstellungs-Verwaltungs-Software**

**Pflichtenheft**

Peter Humer & Daniel Kasper

Projekt WIFI OÖ GmbH 2021

[3884 Software Engineering und Projekt in C#](https://www.wifi-ooe.at/kurs/3884-software-engineering-und-projekt-in-c)

Inhaltsverzeichnis

[1. Benutzerdiagramm 2](#_Toc75846733)

[2. Anwendungsfälle 2](#_Toc75846734)

[3. Szenarien je Anwendungsfall 3](#_Toc75846735)

[4. Das Ablaufdiagramm 4](#_Toc75846736)

[5. Prototyp der Oberfläche 6](#_Toc75846737)

[6. Architektur der Anwendung 9](#_Toc75846738)

[7. Aufgabenassistent 10](#_Toc75846739)

[8. Datenbank 12](#_Toc75846740)

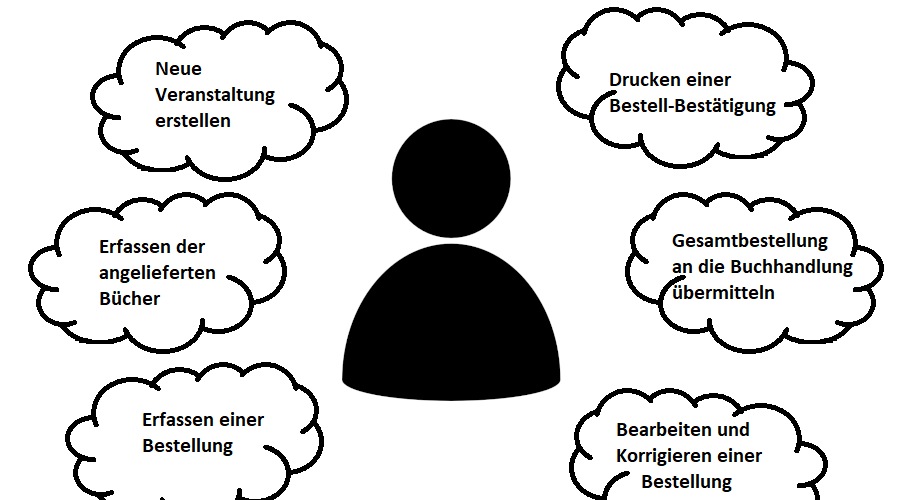
[9. Finalisierte Oberfläche 13](#_Toc75846741)

[10. Codeverteilung 17](#_Toc75846742)

## 1. Benutzerdiagramm



## 2. Anwendungsfälle



## 3. Szenarien je Anwendungsfall

|  |  |
| --- | --- |
| Anwendungsfall | Beschreibung |
| Neue Veranstaltung erstellen | Aus dem Vorjahr vorhandene Daten werden gelöscht und es wird mit einer leeren Datenbank begonnen. |
| Erfassen der angelieferten Bücher | Die von der Buchhandlung angelieferten Bücher werden in der Datenbank erfasst.  Dabei werden die Buchgruppen extra erfasst. |
| Erfassen einer Bestellung | Die Bestellung eines Veranstaltungsbesuchers in der Datenbank erfassen. Alle ausgewählten Bücher werden samt benötigter Anzahl in der Datenbank gespeichert. |
| Drucken einer Bestellbestätigung | Dem Besucher der Veranstaltung wird eine Bestellbestätigung ausgedruckt. |
| Gesamtbestellung an die  Buchhandlung übermitteln | Nach der Veranstaltung wird der Buchhandlung eine ausgedruckte Liste mitgegeben, auf der für alle Bücher die benötigte Gesamtanzahl erfasst ist. |
| Bearbeiten und Korrigieren einer  Bestellung | Falls nach der Veranstaltung von der Buchhandlung nicht alle Bücher in der benötigten Anzahl geliefert werden, muss die Bestellung korrigiert werden und dem Besteller eine neue Bestätigung gedruckt werden. |

## 4. Das Ablaufdiagramm

Starten der Anwendung

* Wiederherstellen der alten Fensterposition
* Anzeigen des Startbildschirms

Klick auf Button „Software starten“

* Software prüft das aktuelle Stadium der Veranstaltung  
  und zeigt dementsprechend die Menüpunkte an

Stadium „Vorbereitung“

* Menüpunkt „Kategorieverwaltung“
  + Erfassen von Buchgruppen
  + Löschen von Buchgruppen
* Menüpunkt „Buchverwaltung“
  + Erfassen von Büchern
  + Import und Export von Büchern via .csv-File
* Menüpunkt „Veranstaltungserstellung“
  + Erfassen des Ortes und der Daten

der Veranstaltung zum Starten der Ausstellung

Stadium „Veranstaltung“

* Menüpunkt „Ausstellung“
  + Anzeige aller verfügbaren Bücher
  + Button zum Bestellen eines Buches
* Menüpunkt „Aktuelle Bestellung“
  + Auflistung der aus der Ausstellung  
    ausgewählten Bücher
  + Erfassen der Daten des Bestellers
* Menüpunkt „Bestellungen“
  + Anzeigen aller erfassten Bestellungen
  + Bearbeiten einer erfassten Bestellung
    - Ändern der Anzahl der bestellten Bücher
    - Ändern der Daten des Bestellers
* Menüpunkt „Ausstellungsabschluss“
  + Gesamtbestellliste als PDF speichern
  + Bestellbestätigungen als PDF speichern
  + Beenden der Ausstellung

Stadium „Lieferverwaltung“

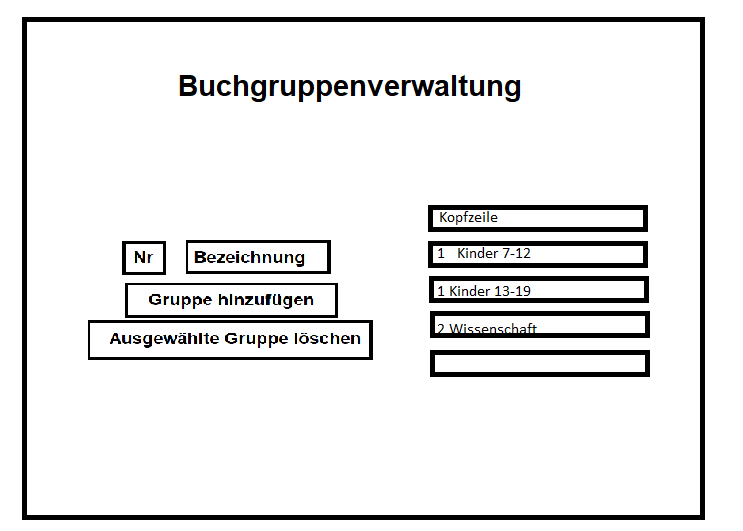
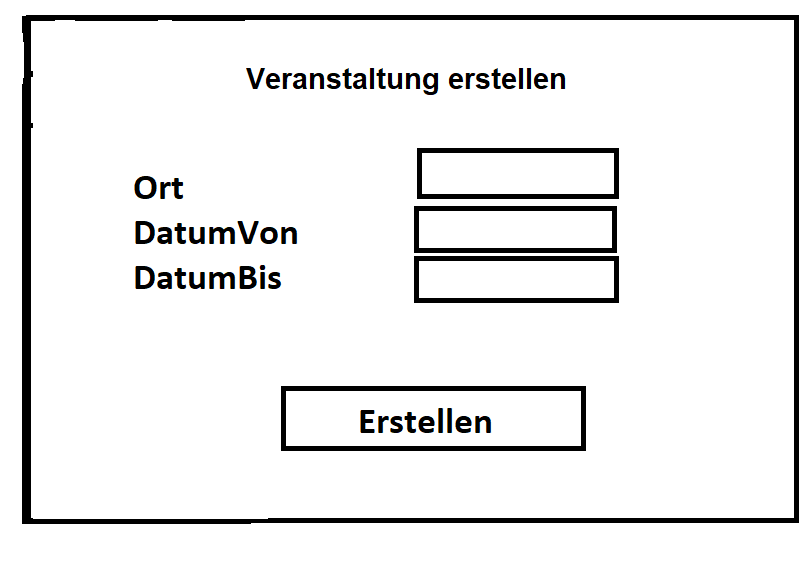
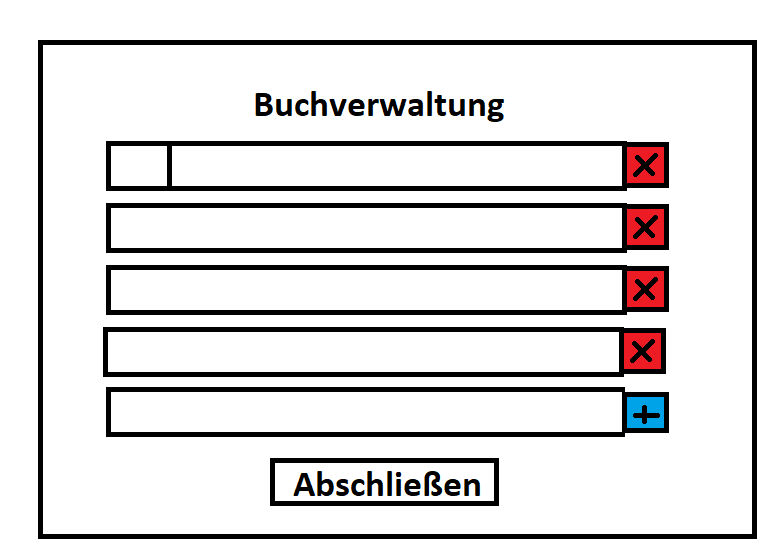
* Prüfen, ob zu jeder Bestellung alle Bücher vorhanden sind
* Abändern der Anzahl eines Buches, falls weniger geliefert wurden

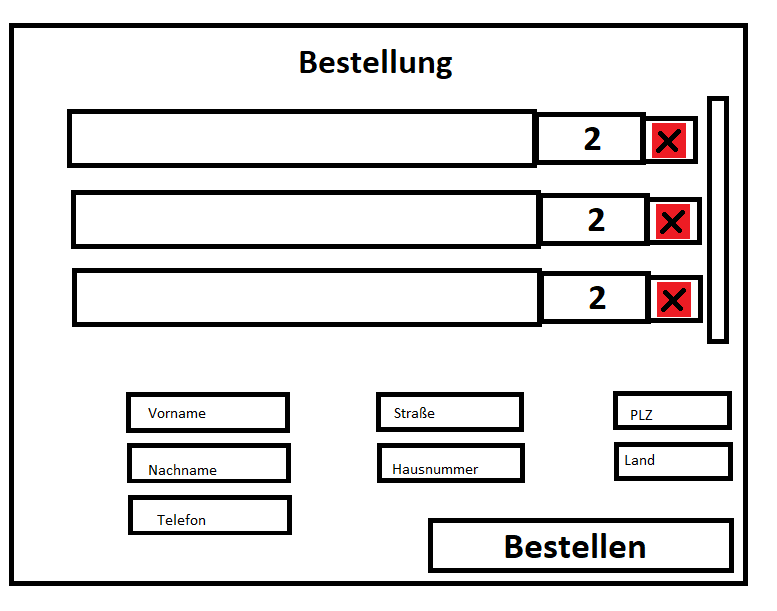
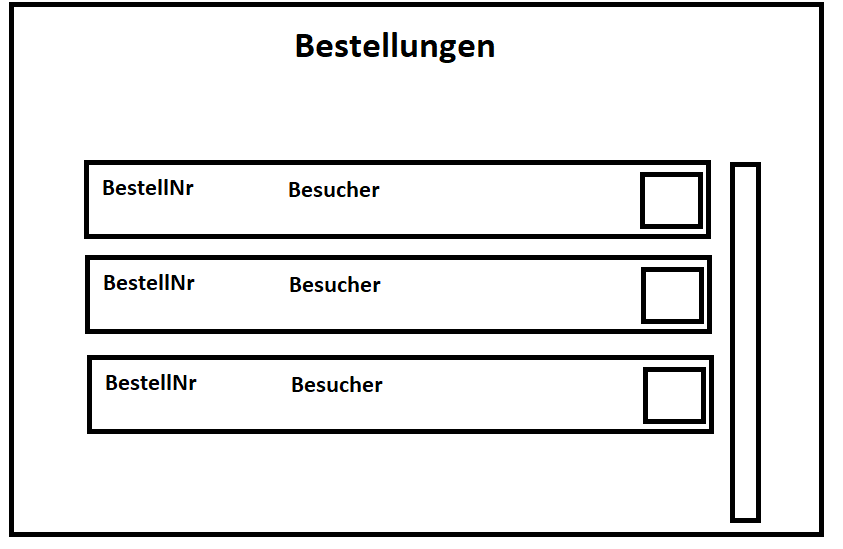
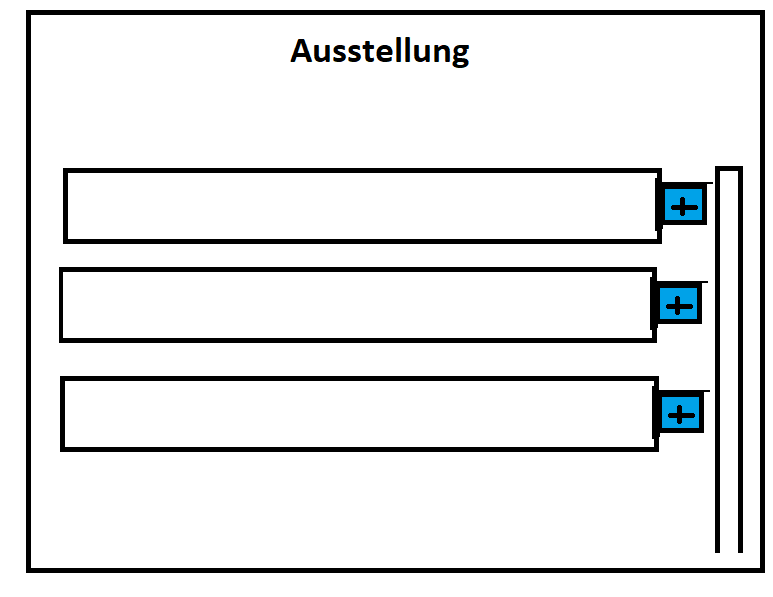
Stadium „Abholungsverwaltung“

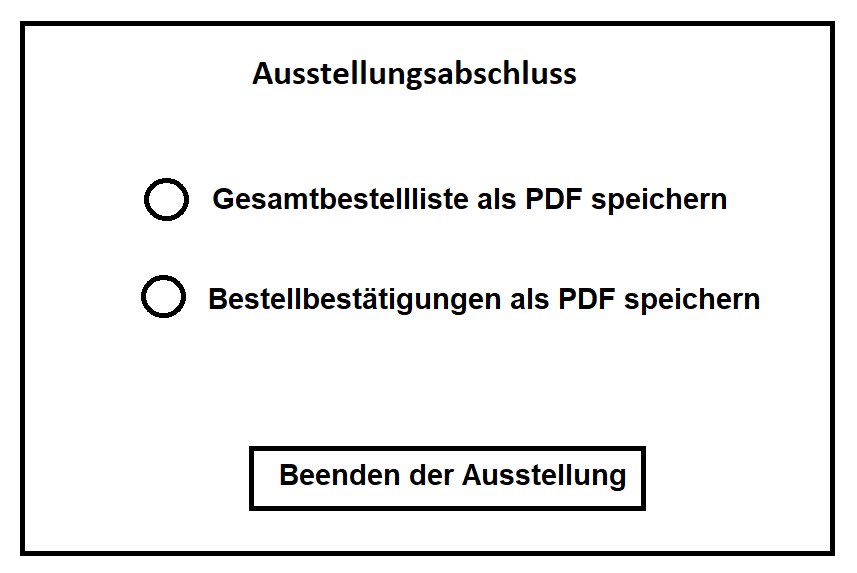
* Kontrollmöglichkeit, ob eine bestellte Lieferung abgeholt wurde

## 5. Prototyp der Oberfläche

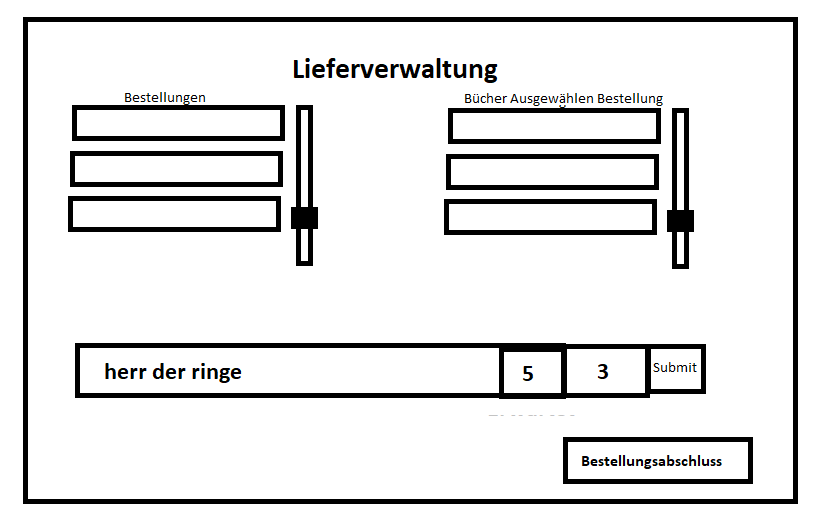
5.1 Vorbereitungs-Stadium-Masken:

****5.2 Veranstaltungs-Stadium-Masken:

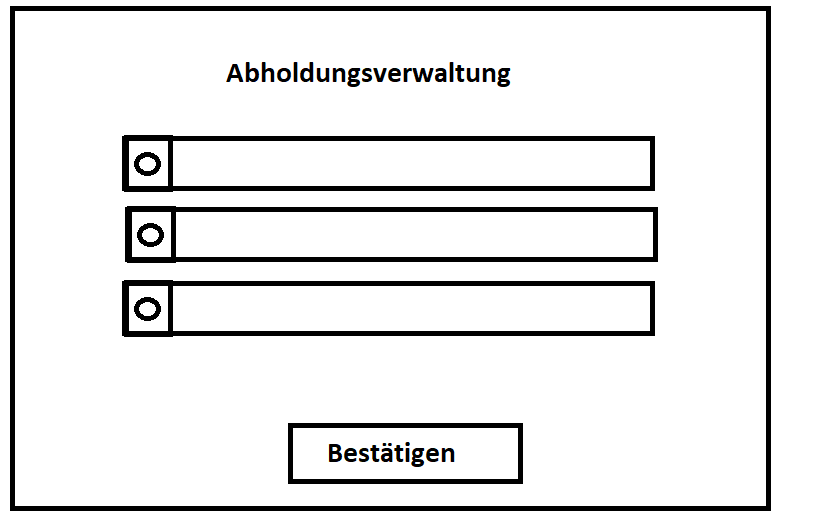




5.3 Lieferung-Stadium-Masken:

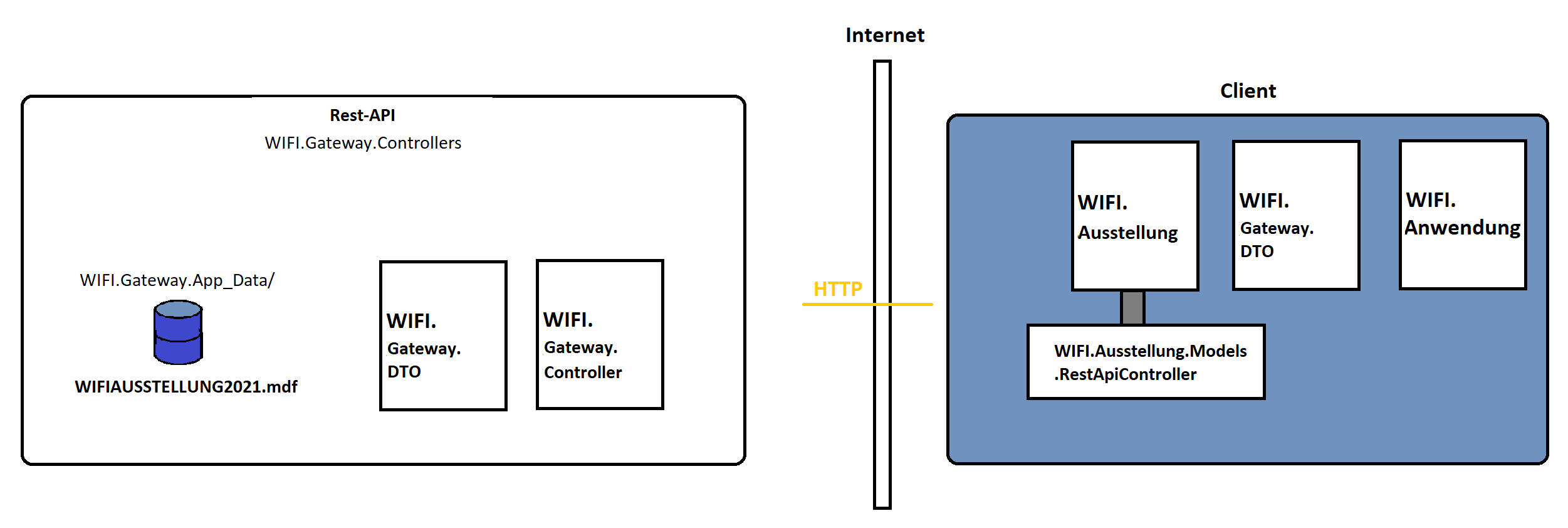


5.4 Abholung-Stadium-Masken:



## 6. Architektur der Anwendung

6.1 Grundaufbau:



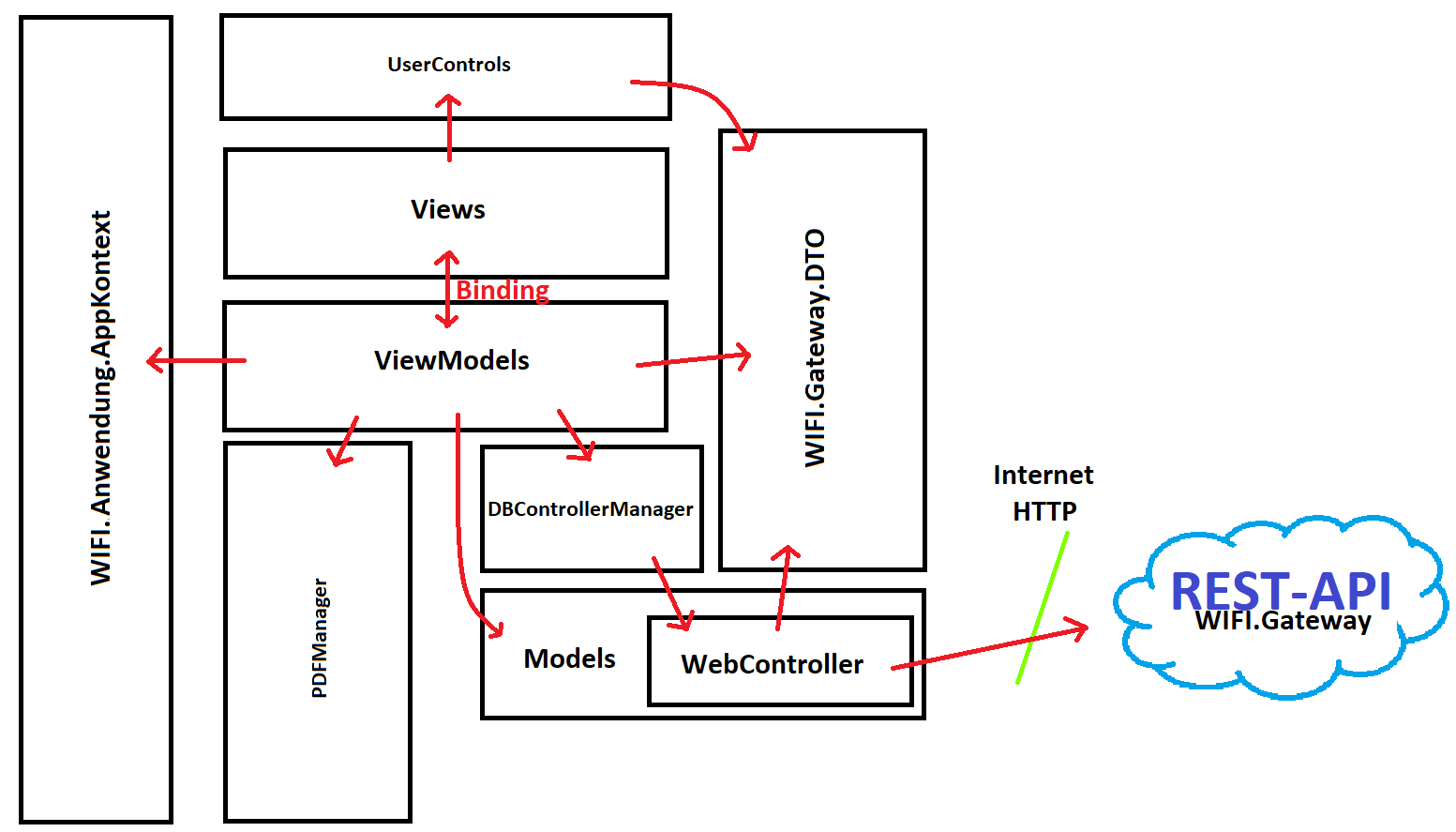
6.2 Client-Software:

Die Anwendung ist nach der MVVM-Architektur (Model-View-ViewModel) entworfen und kommuniziert mit einer REST-API über XML-Format.

Die ViewModels sind unterteilt in:

* Anwendung ( Kern-ViewModel)
  + AufgabenManager
  + VeranstaltungsManager
  + AusstellungsManager
  + BuchManager
  + LieferungsManager
  + AbholungsManager

Die REST-API-WebController zum Kommunizieren mit der Datenbank werden über einen Statischen DBControllerManager unabhängig von der Infrastruktur zur Verfügung gestellt.

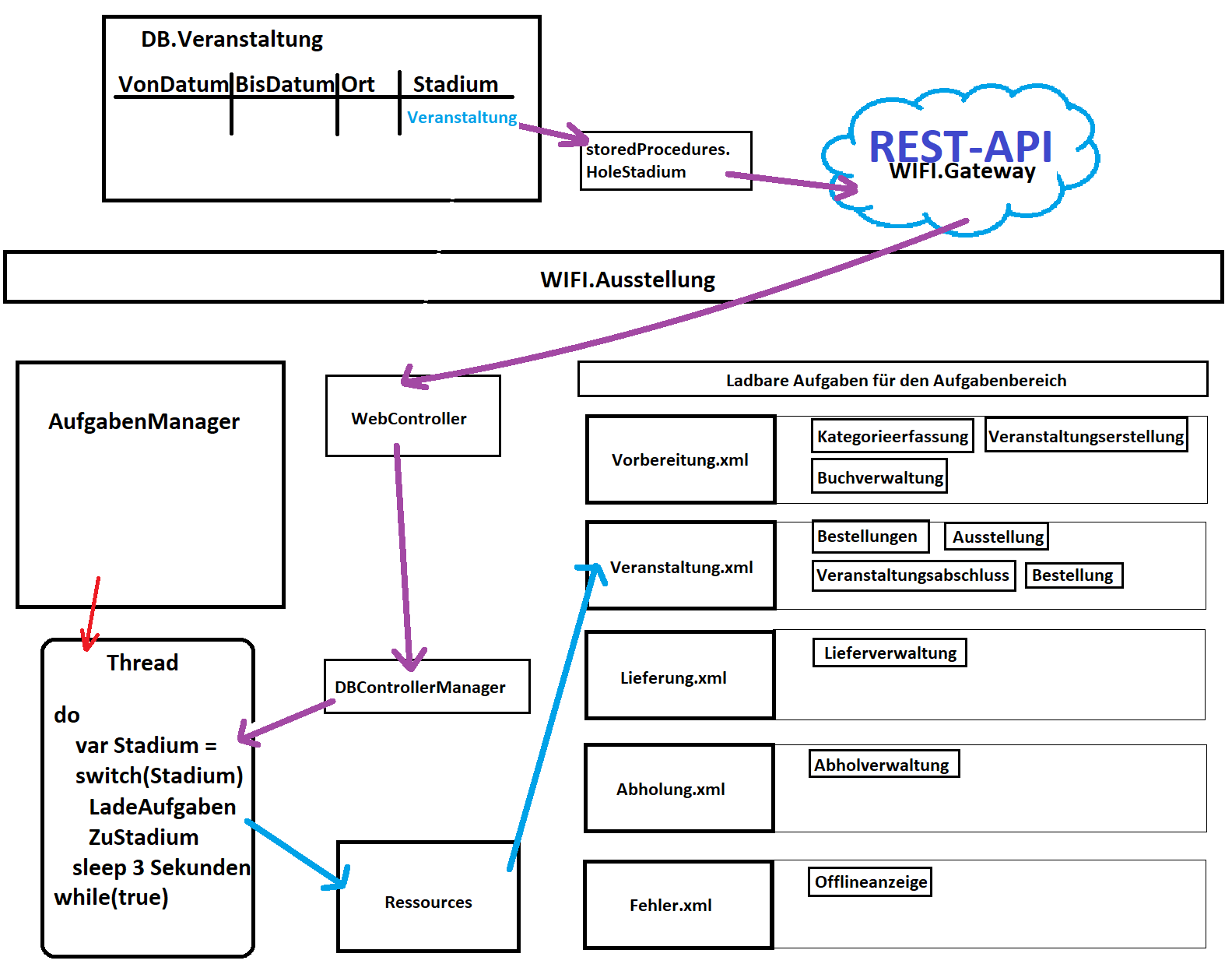
****

## 7. Aufgabenassistent

Der Aufgabenassistent lädt in einem gewissen Takt die verfügbaren Ansichten und Verwaltungsmasken zu dem aktuellen Veranstaltungs-Stadium.

Dieses Stadium kann eines der folgenden sein:

* Vorbereitung
* Veranstaltung
* Lieferung
* Abholung



## 8. Datenbank

8.1 Datenbank:

Ursprünglich war eine MySql Datenbank geplant. Da bei der Umsetzung dieser jedoch die Datensicherheit nicht komplett gewährleistet war, musste die Datenbank in der letzten Woche auf eine ASP.NET – REST API mit einer Microsoft SQL-Datenbank migriert werden.

8.2 Struktur der Datenbank:

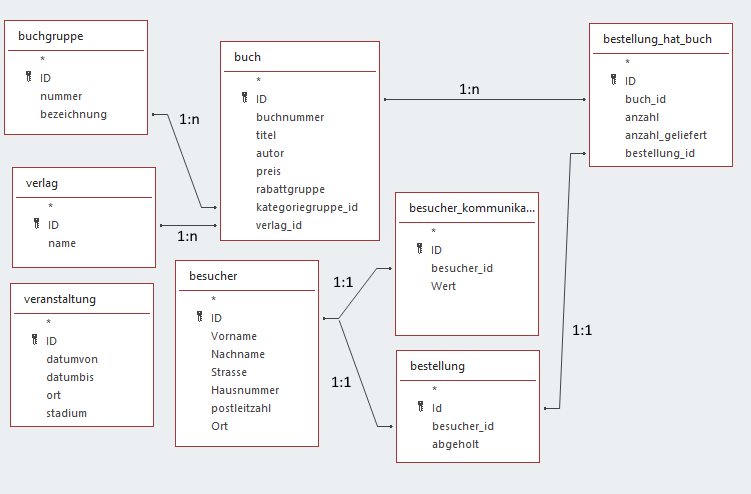
Die Tabelle „veranstaltung“ verwaltet die Veranstaltung an sich.

In der Tabelle „buch“ werden alle Bücher erfasst, die bei der Veranstaltung zur Verfügung stehen.

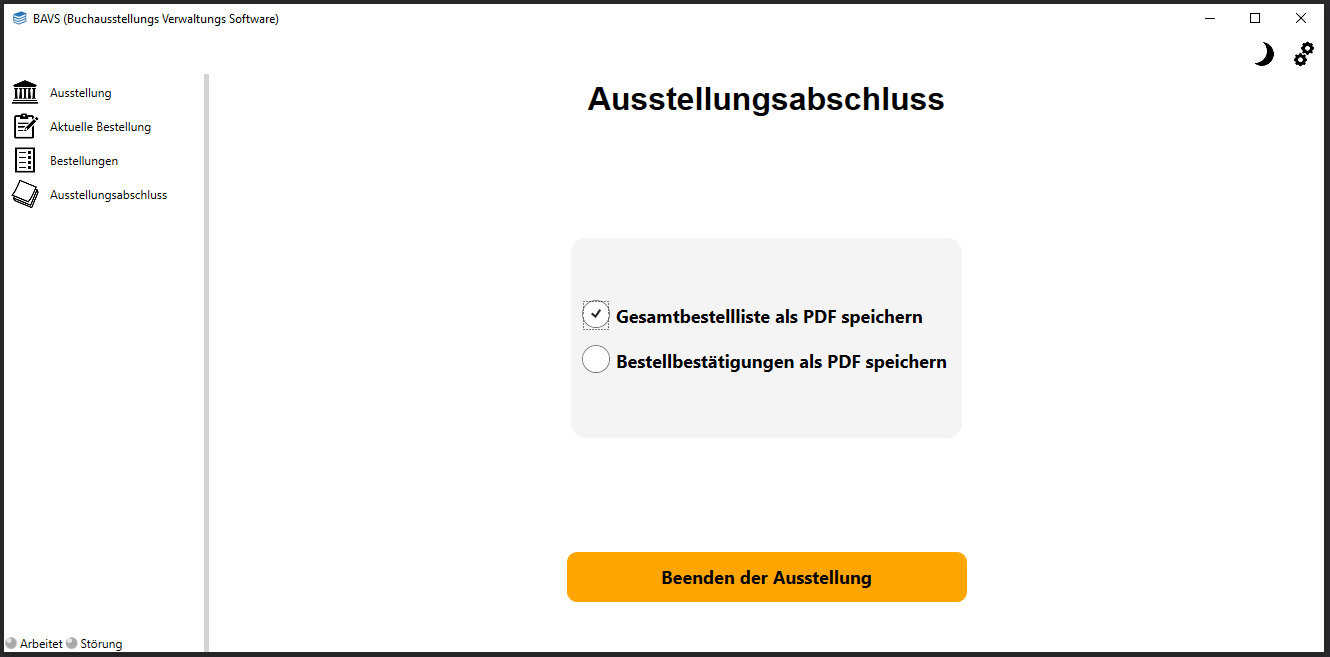
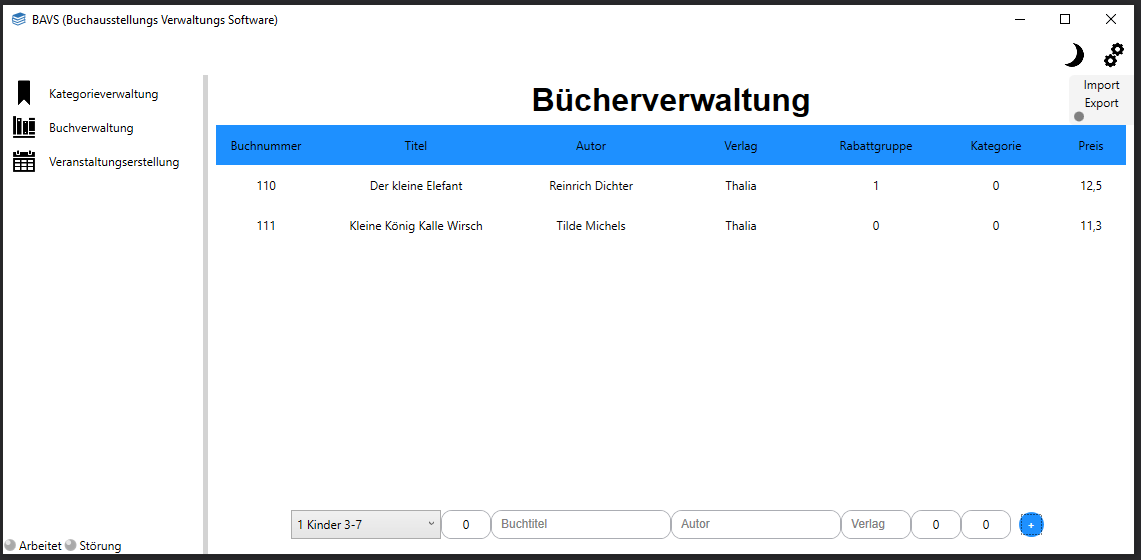
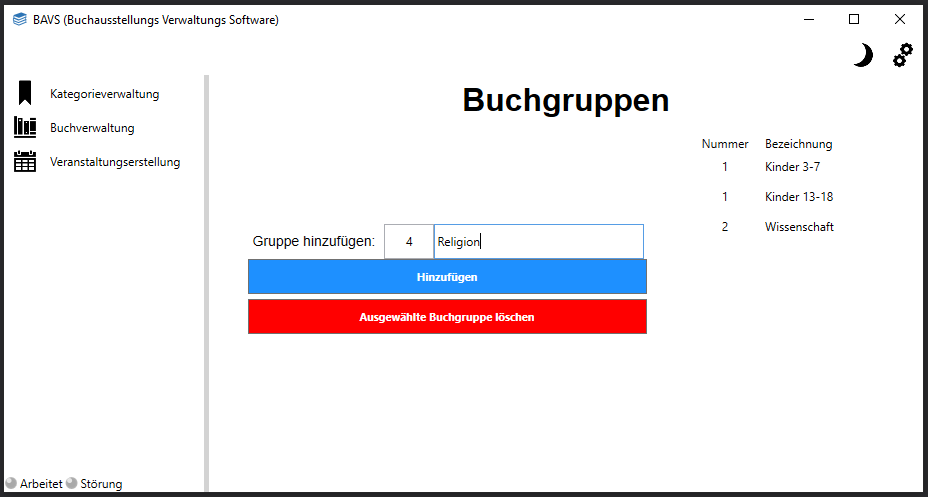
Die Buchgruppe eines Buches ist in der Tabelle „buchgruppe“ erfasst und hat eine Beziehung zur Tabelle buch. Genauso verhält es sich mit der Tabelle „verlag“.

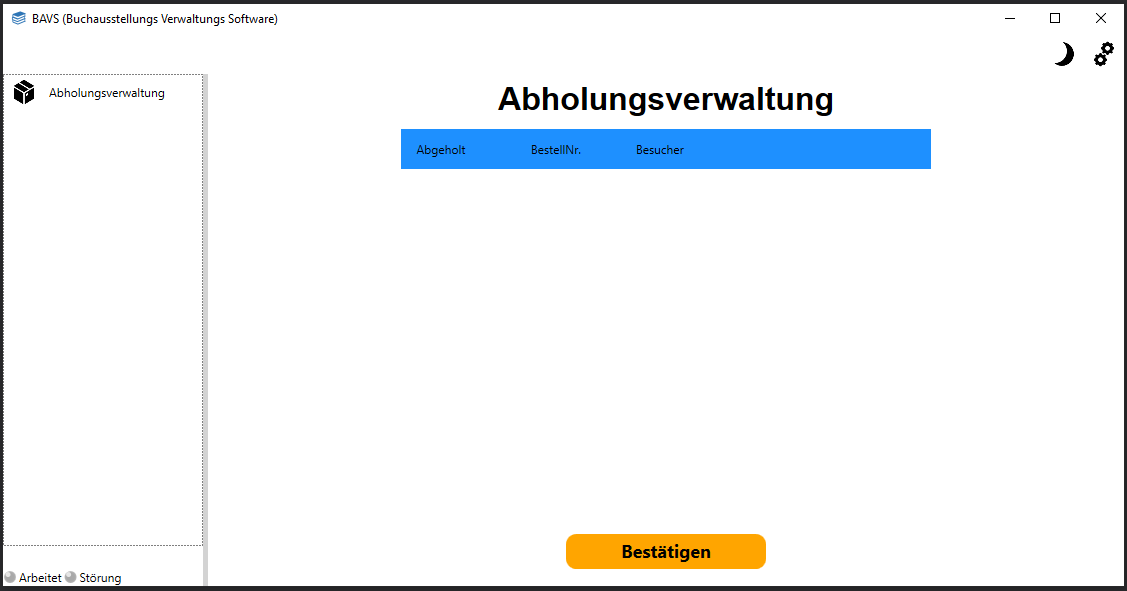
Jeder Besucher, der etwas bestellt, wird in der Tabelle „besucher“ erfasst, die dazugehörigen Kommunikationsdaten in der Tabelle „besucher\_kommunikation“.

Die vom Besucher ausgefüllte Bestellung ist in der Tabelle „bestellung“ abgebildet. Dabei werden die zur Bestellung gehörigen Bücher in der Tabelle „bestellung\_hat\_buch“ abgespeichert.



## 9. Finalisierte Oberfläche





## 10. Codeverteilung

Das gesamte Projekt wurde in einem GitHub-Repository umgesetzt und ist unter

<https://github.com/zenoxart/Buchausstellung>

jederzeit zu finden.