Programmieren 3 (WIF)

Wintersemester 2015/16

Studienprojekt

**Spezifikation**

|  |  |
| --- | --- |
| **Projekt:** | ***Bauhaus Bar Verwaltung*** |

|  |  |  |
| --- | --- | --- |
| Team: | **Lammbock** | *Tobias Ruckdeschel*  *Robin Wedekind*  *Simon Treutlein* |

Inhalt

[1 Beschreibung 2](#_Toc437289486)

[2 Anwendungsfälle 2](#_Toc437289487)

[2.1 Anwendungsfall 1 2](#_Toc437289488)

[2.2 Anwendungsfall 2 2](#_Toc437289489)

[2.3 Anwendungsfall 3 3](#_Toc437289490)

[2.4 Anwendungsfall 4 3](#_Toc437289491)

[3 Masken 4](#_Toc437289493)

[3.1 „Willkommens“ Tab 4](#_Toc437289494)

[3.1.1 „Bestand“ PopUp 5](#_Toc437289495)

[3.2 „Getränke“ Tab 6](#_Toc437289496)

[3.3 „Snack“ Tab 7](#_Toc437289497)

[3.4 „Hinzufügen“ Tab 8](#_Toc437289498)

[3.5 „Entfernen“ Tab 9](#_Toc437289499)

[4 Fachliches Datenmodell 10](#_Toc437289500)

[5 Komponenten Struktur 11](#_Toc437289501)

[5.1 Beschreibung der Komponenten 11](#_Toc437289502)

[5.1.1 Getränke 11](#_Toc437289503)

[5.1.2 Snacks 11](#_Toc437289504)

[5.1.3 Lager 11](#_Toc437289505)

[6 Validierung 11](#_Toc437289506)

[6.1 Testfall 1 Mindestbestand überprüfen 11](#_Toc437289507)

[6.2 Testfall 2 Testentnahme 11](#_Toc437289508)

[6.3 Testfall 3: Objekt hinzufügen 12](#_Toc437289509)

[7 Aufteilung der Aufgaben 12](#_Toc437289510)

# Beschreibung

Bei der Anwendung handelt es sich um ein Verwaltungssystem für ein Lager einer Bar. Anwender sind kleinere Bars, in unserem Fall ist der Anwender eine Bar in Vaterstetten. Die Idee ist, dass man durch die Anwendung eine bessere Übersicht über den aktuellen Lagerbestand bekommt.

# Anwendungsfälle

Der Benutzer kann in der Anwendung sein Lager verwalten, das heißt er kann dem Lager neue Getränke hinzufügen und entfernen. Er kann prüfen, wieviel Getränke einer speziellen Sorte noch lagernd sind und

## Anwendungsfall 1

Der Benutzer prüft wieviel Getränke auf Lager sind.

* Anstoß des Anwendungsfalls:

Dieser Fall wird manuell durch den Benutzer ausgelöst.

* Daten:

Die Daten werden gelesen.

* Ablaufbeschreibung:

Der Benutzer öffnet die Anwendung und klickt auf den Button um zur Lagerübersicht zu kommen.

## Anwendungsfall 2

Die Anwendung erinnert den Benutzer, dass von einem Getränk wenig auf Lager ist und nachgekauft werden sollte.

* Anstoß des Anwendungsfalls:

Dieser Fall wird automatisch beim Öffnen der Anwendung ausgelöst.

* Daten:

Die Daten werden gelesen, geprüft und anschließend das Ergebnis ausgegeben.

* Ablaufbeschreibung:

Der Benutzer öffnet die Anwendung und es kommt, sofern nötig, eine Meldung, dass zu wenig von einem bestimmten oder mehreren Getränken auf Lager ist.

## Anwendungsfall 3

Der Benutzer fügt dem Lager Getränke hinzu.

* Anstoß des Anwendungsfalls:

Dieser Fall wird manuell durch den Benutzer ausgelöst.

* Daten:

Die Daten werden geschrieben.

* Ablaufbeschreibung:

Der Benutzer öffnet die Anwendung und klickt auf den Button um zur Lagerverwaltung zu kommen. Danach klickt er auf den Button um neue Getränke hinzuzufügen und gibt die notwendigen Daten ein. Nachdem er auf speichern geklickt hat werden die Daten in das Lager aufgenommen.

## Anwendungsfall 4

Der Benutzer löscht Getränke aus dem Lager.

* Anstoß des Anwendungsfalls:

Dieser Fall wird manuell durch den Benutzer ausgelöst.

* Daten:

Die Daten werden geschrieben.

* Ablaufbeschreibung:

Der Benutzer öffnet die Anwendung und klickt auf den Button um zur Lagerverwaltung zu kommen. Danach klickt er auf den Button um Getränke zu entfernen und gibt die notwendigen Daten ein. Nachdem er auf speichern geklickt hat werden die Daten in das Lager aufgenommen.

# Masken

Die GUI besteht aus einem Hauptfenster. Der Benutzer navigiert durch die verschiedenen „Menüs“ mithilfe von Tabs, ähnlich denen von z.B. Chrome.

Das Programm-Fenster öffnet sich in der Bildschirmmitte.

## „Willkommens“ Tab



Abbildung – Beispielmaske „Willkommen“

Der „Willkommen“ Tab ist der erste den der Benutzer sieht.

Mit ihm wird gleichzeitig, automatisch die Überprüfung auf den Mindestbestand gestartet.  
(siehe 3.1.1 „Bestand“ PopUp)

Hier gibt es keine weiteren Interaktionsmöglichkeiten. Dieser Tab dient lediglich als „Start-“ oder „Begrüßungsbildschirm“.

### „Bestand“ PopUp



Abbildung – Beispielmaske „PopUp“

Das PopUp öffnet sich nur wenn der Mindestbestand eines Produktes unterschritten wurde, und zeigt das entsprechende Produkt/Produkte an.

Im PopUp gibt es keine Interaktionsmöglichkeiten.

Das PopUp soll sich am unteren Rand der Anwendung öffnen.

## „Getränke“ Tab

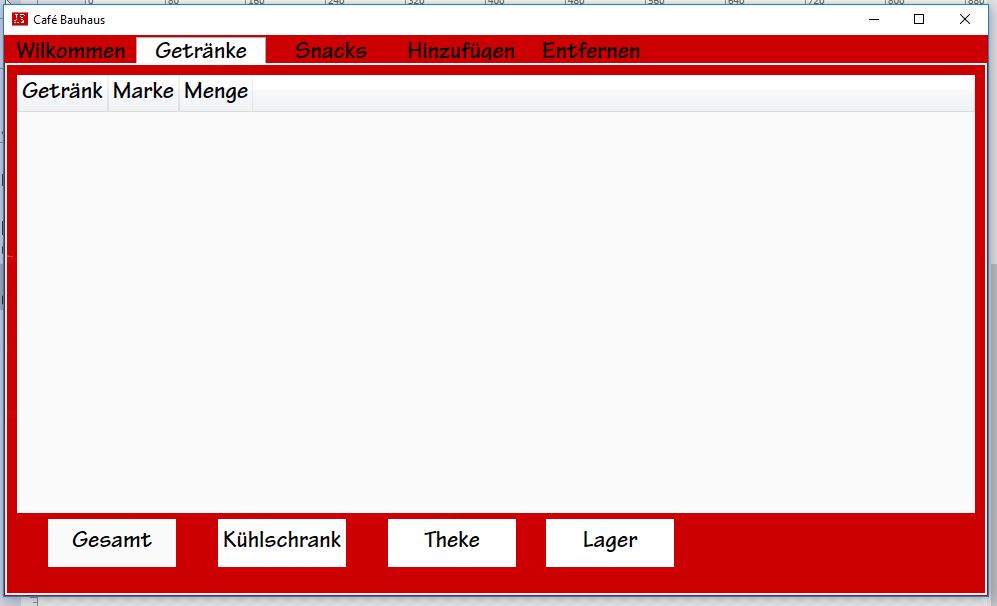


Abbildung 3 – Beispielmaske „Getränke“

Im „Getränke“ Tab befindet sich die Tabelle für die Getränke Daten.

Standardmäßig ist die Tabelle leer.

Per Mausklick auf die Buttons *„Gesamt“*, *„Kühlschrank“*, *„Theke“* und *„Lager“* kann man hier die jeweiligen Getränke Bestände anzeigen lassen.

Der Datenzugriff erfolgt rein lesend.

## „Snack“ Tab



Abbildung 4– Beispielmaske „Snacks“

Im „Snack“ Tab befindet sich die Tabelle für die Snack Daten.

Standardmäßig ist die Tabelle leer.

Per Mausklick auf die Buttons *„Süß“* und *„Salzig“* kann man hier die jeweiligen Snack Bestände anzeigen lassen.

Der Datenzugriff erfolgt rein lesend.

## „Hinzufügen“ Tab



Abbildung 5 – Beispielmaske „Hinzufügen“

Im Tab „Hinzufügen“ können neue Getränke oder Snacks zur jeweiligen Datenbank hinzugefügt werden.

Über entsprechende Drop-Down Auswahl können Datensätze bei Betätigung des „Buchen“ Buttons in die jeweilige Datenbank übertragen werden.

Der Datenzugriff erfolgt hier rein schreibend.

## „Entfernen“ Tab



Abbildung 6 – Beispielmaske „Entfernen“

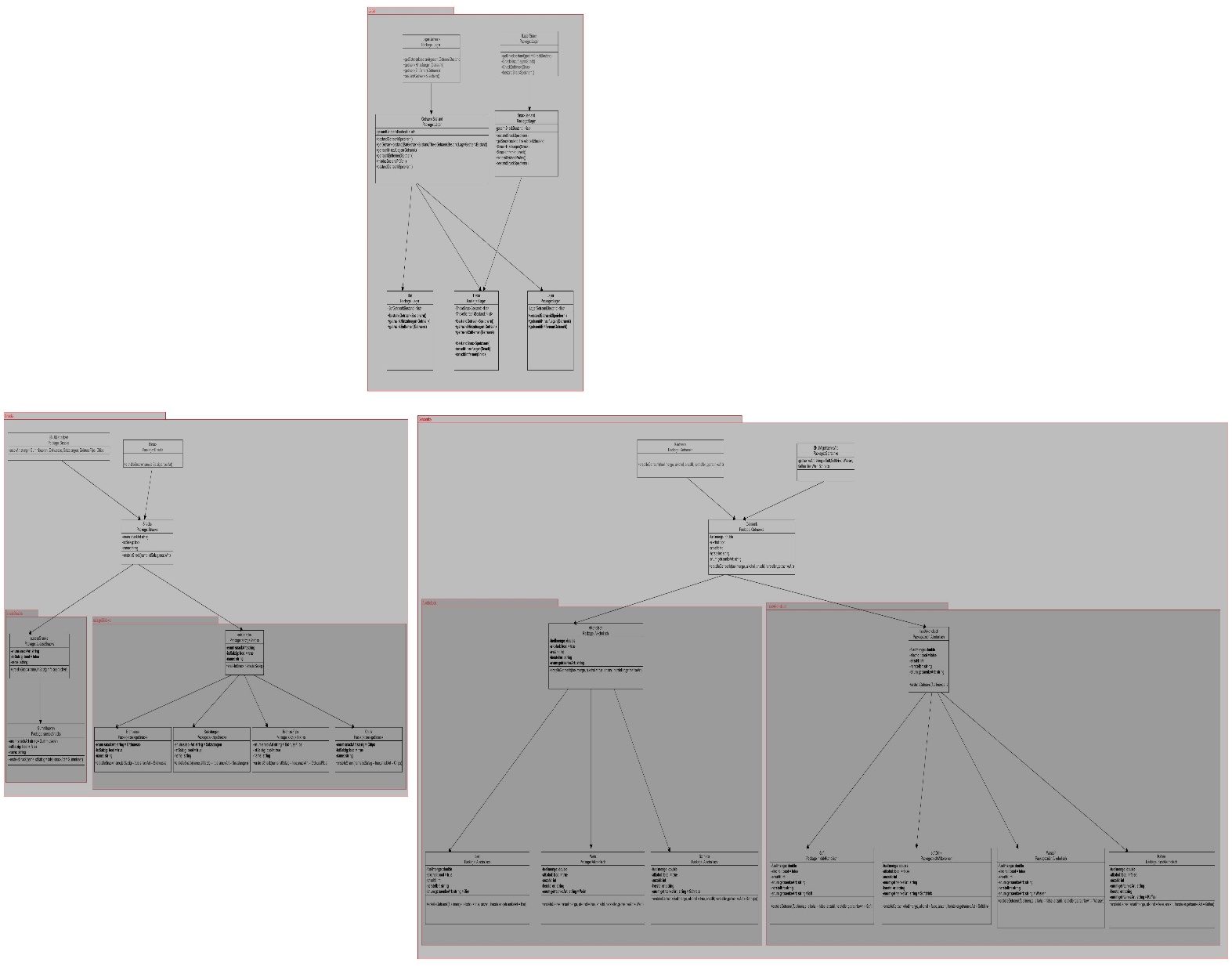
Im Tab „Entfernen“ können neue Getränke oder Snacks aus der jeweiligen Datenbank entfernt werden.

Über entsprechende Drop-Down Auswahl können Datensätze bei Betätigung des „Entfernen“ Buttons aus der jeweilige Datenbank gelöscht werden.

Der Datenzugriff erfolgt hier rein schreibend (entfernend).

# Fachliches Datenmodell

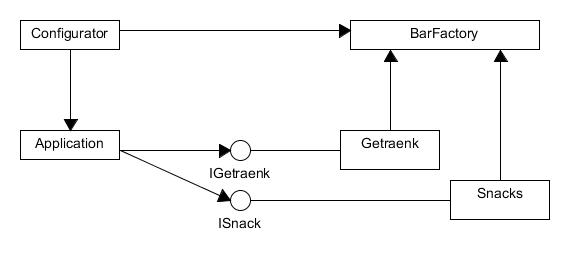
UML Diagramm:

****

# Komponenten Struktur

Beschreiben Sie hier die Komponentenstruktur ihrer Anwendung:

* Beschreibung der Komponenten (Aufgabe, Außensicht, Innensicht)
* Beschreibung der Schnittstellen
  + Methoden mit exakter Signatur
  + Datentypen der Schnittstelle
* Machen sie ein Architekturbild aus dem die Komponentenstruktur und die Schichtenarchitektur hervorgehen.



## Beschreibung der Komponenten

### Getränke

Aufgabe der Getränkekomponente ist Darstellung der eingelagerten Getränke. Als Schnittstelle ist die Schnittstelle IGetränke vorgesehen, welche die verschiedenen benötigten Methoden beinhaltet. Implementiert werden diese Methoden in den einzelnen Klassen.

### Snacks

Wie auch in der Getränkekomponente ist die Klasse Snacks für die Haltung der eingelagerten Getränke zuständig. Ebenfalls ist hier eine Schnittstelle – ISnacks- vorgesehen.

### Lager

Das Lager stellt die Datenbank der eingelagerten Getränke und Snacks dar. Es dient zum abspeichern und Lesen der einzelnen Objekte. Hier wird eine Schnittstelle ILagerSnacks sowie ILagerGetraenk vorgesehen, welche Methoden zur Speicherung und zum Lesen der eingelagerten Getränke und Snacks als Funktion hat.

# Validierung

## Testfall 1 Mindestbestand überprüfen

Es wird ein Mindestbestand festgelegt, welcher zu überprüfen ist. Angenommen wird, dass der Mindestbestand unterschritten wird. Erwartet wird bei erfolgreichen Durchlauf eine Fehlermeldung, welche eine Unterschreitung des Mindestbestandes anzeigt.

## Testfall 2 Testentnahme

Bei diesem Testfall soll überprüft werden, ob bei der Entnahme eines Objektes der Bestand korrigiert wird. Die Testklasse simuliert eine Entnahme eines Getränkes. Falls der Testfall erfolgreich durchgeführt wird, muss der Bestand ein Objekt weniger betragen.

## Testfall 3: Objekt hinzufügen

Es wird ein Getränk/Snack hinzugefügt, ist der Test erfolgreich wird der Bestand um ein Objekt mehr korrigiert.

# Aufteilung der Aufgaben

**Tobias Ruckdeschel**

* GUI – Design + Funktionalitäten
* Startmethode

**Robin Wedekind**

* Datenbank
* Mindestbestand

**Simon Treutlein**

* Implementierung