

# Qualitätssicherung (QS) VU: Beispiel „Restaurant“

Im Rahmen des Übungsteils der VU Qualitätssicherung spezifiziert dieses Dokument Anforderungen an ein System und dessen Modelle als Basis für Reviews.

- Software-Requirement Review
- Preliminary-Design Review

Die Ergebnisse der Reviews sind im Dokument „Leitfaden“ aufzuzeichnen.

## 1. Beschreibung

Für ein Restaurant soll ein Software-System entwickelt werden, das die Prozesse des Restaurantpersonals effektiv und effizient unterstützt. Das System soll zur Lagerverwaltung, Bestellung und Kundenabrechnung, sowie zur Verwaltung des Personals und Kunden verwendet werden.

Kurzfristige Ziele des hier zu modellierenden Systems sind die reibungslose Reservierung von Speisen und Tischen, die Abwicklung von Bestellungen, das Erstellen von Tagesplänen für den Einkauf von Zutaten, die Buchführung im Lager über die vorhandenen Zutaten, und schließlich die Abrechnung der erbrachten Leistungen mit den Kunden. Mittelfristige Ziele sind die automatische Berechnung von internen Kosten für Einkauf, Lager und Zubereitung. Langfristige Ziele sind das Gewinnen von Kunden durch Anbieten von zugkräftigen Leistungen, die Senkung der Kosten für den Einkauf sowie die Verringerung des Abfalls, bestehend aus abgelaufenen Zutaten im Lager, die nicht mehr verwendet werden können.

Im Zuge einer Bestellung stellt der Kunde aus den Menüvorschlägen oder einzelnen Speisen der Speisekarte eine Auswahl zusammen. Dabei ist anzugeben, wann das Essen stattfinden soll und ob es im Restaurant gegessen oder mitgenommen wird. Zu einer Bestellung über €250.- ist eine Anzahlung von 10% zu leisten. Zu jeder aufgenommenen Bestellung erhält der Kunde eine Bestellnummer, über die er die Bestellung stornieren kann. Eine geleistete Anzahlung verfällt bei Stornierung.

Die Speisen der Speisekarte haben jeweils eine eindeutige Bezeichnung und einen Verkaufspreis, manche Speisen sind nicht immer bestellbar. Ein Menü ist eine Auswahl von Speisen der Speisekarte zu einem günstigeren Preis und steht selbst nicht auf der Speisekarte. Meist ist ein Menü nur in einem bestimmten Zeitraum zu haben und hat eine thematische Bezeichnung, etwa 'Variationen von Kürbissen' erhältlich im Oktober und November.

Intern gibt es zu jeder Speise mindestens ein Rezept, das neben dem Zeitaufwand, den notwendigen Werkzeugen und der Anzahl daraus entstehender Portionen auch die Zutaten mit der benötigten Menge auflistet. Ein kompliziertes Rezept kann aus einfacheren Rezepten bestehen (Etwa steht in einem Rezept über 'Altwiener Erdäpfeltopf' "... Man bereite eine Einbrenn vor...").

Bezeichnung	Menge	Einheit	Preis/Einh	Preis
Salat grün	25	Stk	0.40	10.00
Kartoffel Sigma	15	kg	0.36	5.40
Vollmilch	120	Liter	0.70	840.00
Schlagobers	90	Liter	0.95	88,00
Semmel	250	Stk	0.08	22.50
...			<i>Summe</i>	<i>965.90</i>

Abb. 1: Auszug aus einer Einkaufsabrechnung am 12.10.2006.

Einmal pro Tag geht der Einkäufer auf den Markt, um die Zutaten für den aktuellen Tag zu besorgen. Dabei verwendet er eine Einkaufsliste, die aufgrund der vorgemerkten Bestellungen

gen und den im Restaurant auf Lager liegenden Zutaten erstellt wird. Im Lager liegt ein Journal auf in das der Buchhalter an jedem Abend nach Geschäftsschluss für jede Zutat die auf QS VU Aufgabenstellung\_Restaurant

Lager liegende verwendbare Menge vermerkt sowie die Menge an Abfall, das sind Zutaten die im Lager verdorben sind. Nach jedem Einkauf werden die frischen Zutaten eingelagert. Eine Aufstellung der Menge der gekauften den Buchhalter zur späteren Abrechnung.

Kategorie	Bezeichnung	Einheit	verwendbar	Abfall
Gemüse	Salat grün	Stk.	90	32
Gemüse	Kartoffel Sigma	kg	160	4
Milchprodukte	Vollmilch	Liter	17	23
Milchprodukte	Schlagobers	Liter	7	1
Gebäck	Semmel	Stk.	88	0
	...			

Abb. 2: Auszug aus dem Lagerjournal nach der Inventur der Zutaten am 12.10.2006.

Für die Zubereitung der Speisen erhält der Koch jeden Morgen einen Tagesplan der vorliegenden Bestellungen. Entsprechend werden Speisen vorbereitet und fertig gestellt. Nicht verbrauchte verwendbare Zutaten werden wieder eingelagert, eine Kopie aller Rechnungen über zubereitete Speisen gehen in die Buchhaltung zur Abrechnung. Zu jeder Rechnung für einen Kunden werden die Endsumme und der verantwortliche Koch vermerkt.

Im Zuge der täglichen Abrechnung wird überprüft, ob alle Bestellungen wie vorgesehen erfüllt wurden. Noch nicht bezahlte Rechnungen werden ausgedruckt. Für jeden Kunden gibt es einen Kontostand sowie eine Kontaktadresse. Am Ende der Woche wird aus den Kosten für Einkäufe und den Einnahmen eine vorläufige Bilanz erstellt.

*Integritätsbedingungen:* Der Menge eingekaufter Zutaten werden die hergestellten Speisen sowie die Menge an verdorbenen Zutaten gegenübergestellt. Dabei wird überprüft, ob der Lagerstand am Ende der Vorwoche plus die Menge der Einkäufe in dieser Woche auch tatsächlich die Menge der für Speisen verbrauchten Zutaten plus den aktuellen Lagerstand plus die Summe der Abfälle dieser Woche ergibt.

*Auswertungen:* 1. Das neue System soll anhand der Vorbestellungen von Speisen einen Vorschlag über die benötigte Menge an Zutaten machen. 2. Für jede Speise soll das System entscheiden können, ob sie mit den vorhandenen Zutaten gekocht werden kann oder nicht. 3. Für Marketingzwecke soll es möglich sein, pro Kunde über einen bestimmten Zeitraum die Summe der Bestellungen bzw. bezahlten Rechnungen zu ermitteln.

## Anforderungen (unvollständig)

Die Anforderungen wurden aus der Beschreibung mit Hilfe des Kunden erstellt. Die Darstellung als Liste unterstützt eine systematische Bearbeitung besser als Prosatext.

1. Das System soll die Eingabe und Ausgabe von zwei verschiedenen Arten von Belegen, Rechnung und Verkauf, unterstützen.
2. Zu jeder Rechnung werden die Endsumme, Datum und der verantwortliche Kellner vermerkt.
3. Zu jedem Einkauf wird die Endsumme, Datum und der verantwortliche Einkäufer vermerkt.
4. Die Lagerverwaltung besteht aus einem Journal welches täglich von einem Buchhalter aktualisiert wird (Wochenbilanz). Dabei werden die Daten der Rechnungen und des Einkaufs benötigt. Das System soll das automatische drucken von verlorenen Rechnungen bei der Aktualisierung unterstützen.
5. Wöchentlich muss eine Wochenbilanz des Lagerjournals erstellt werden, bei dieser Aktion soll ein Integritätscheck durchgeführt werden, es muss gelten:
  - Lagerstand + Einkäufe = Lagerstand + verbrauchte Zutaten + eingekaufte Zutaten
  - Im Fehlerfall ist eine Korrekturbuchung zu erstellen (Erklärung für Differenz).
6. Ein Kunde soll zwei Bestellungen auf einmal bezahlen und genau eine Rechnung bekommen können.
7. Für jede Speise soll das System entscheiden können, ob sie mit den vorhandenen Zutaten gekocht werden kann oder nicht.
8. Aufgrund der Vorbestellungen für Speisen, soll das System eine Einkaufsliste für Speisen generieren. Es soll dem Einkäufer möglich sein diesen Vorschlag zu verändern und anschließend zu drucken.
9. Außerdem soll es dem Verkäufer auch möglich sein die Kundenstatistiken einzusehen um ihm bei dem Einkauf zu unterstützen.
10. Jede Speise kann jederzeit vom Kellner bestellt und auch wieder storniert werden. Optional können Speisen vom Gast auch mitgenommen werden. Es gibt keinen Zustelldienst.
11. Das gesamte System verwendet ausschließlich Euro Währung, dabei sollte ein Geldbetrag immer als die kleinste ganze Einheit, also Cent, gespeichert werden.
12. Bei einer Bestellung von über 500 Euro soll eine Anzahlung von 10% erfolgen.
13. Alle Speisen können als Teil eines Menüs zusammengestellt und gespeichert werden.
14. Es kann auch vorkommen dass eine Speise aus nur einer Zutat besteht, z.B. ein Apfel. Ein Apfel ist hier sowohl eine Speise und eine Zutat.
15. Das System soll durch einen Touch-Screen mit Handschuhen bedienbar sein, zusätzlich sollte der Text, besonders die Endsummen, von einem Meter weit weg noch lesbar sein.
16. Als Erweiterung soll das System nach Auslieferung die Anbindung von portablen Geräten unterstützen, insbesondere für die Kellner.
17. Jegliche Batch-Jobs welche automatisch durch das System getriggert werden sowie jene welche eine Manuelle Eingabe benötigen, wie etwa das Eintragen von fehlenden Rechnungen, soll nicht mehr als 3 Stunden in Anspruch nehmen; Das System muss zwischen den Arbeitsstunden 6:00 und 00:00 Uhr immer funktionieren und die Systemintegrität muss in der DB gewährleistet sein.

## Anwendungsfälle

Die Funktionalen Anforderungen werden in einem Anwendungsfall Modell dargestellt. Anwendungsfälle sind Textdokumente, nicht Diagramme. Modellierung der Anwendungsfälle bedeutet primär das Modellieren von Szenarien welche die Funktionalität der Anforderungen widerspiegeln.

Die Beschreibungen der Anwendungsfälle stützen sich also auf die Anforderungen und der Projektbeschreibung und werden mit Hilfe eines Diagramms übersichtlich dargestellt. Das Anwendungsfalldiagramm dient auch dazu weitere Anwendungsfälle zu finden und die vorhandene Textuelle Modellierung zu verfeinern.

## Anwendungsfallbeschreibungen

<i>Name</i>	verkaufte Speisen
<i>Primäre Aktoren</i>	Kellner
<i>Sekundäre Aktoren</i>	Einkäufer
<i>Vorbedingungen</i>	Kellner hat sich am System authentifiziert und ist eingeloggt.
<i>Nachbedingungen</i>	Bestellung ist gespeichert, Bestellungsnummer wurde generiert, Koch wurde definiert, Rechnung wurde ausgedruckt
<i>Szenario</i>	<ol style="list-style-type: none"> <li>1 Kunde bestellt eine Speise oder ein Menü</li> <li>2 Kellner erzeugt neue Bestellung und gibt die Bestellung aus Schritt 1 ein <i>Kellner wiederholt die Schritte 1-2 bis der Bestellvorgang fertig ist.</i></li> <li>3 Das System berechnet die Gesamtsumme <i>Extension Point: Maximalsumme überschritten</i> <i>Die Speise wird vom Koch zubereitet und entweder vom Kellner serviert oder zum mitnehmen eingepackt.</i></li> <li>4 Kellner übermittelt dem Kunden die Gesamtsumme und fragt nach der Bezahlung</li> <li>5 System loggt endgültigen Verkauf und aktualisiert die notwendigen Datensätze im Journal des Lagers. <i>Anwendungsfall: drucke Rechnung</i></li> <li>6 Kunde verlässt das Restaurant mit Rechnung und optional mit Speisen oder Menüs.</li> </ol>

<i>Name</i>	drucke Einkaufsliste
<i>Primäre Aktoren</i>	Einkäufer, System
<i>Sekundäre Aktoren</i>	keine
<i>Vorbedingungen</i>	Lagerstand muss auf aktuellen Stand sein.
<i>Nachbedingungen</i>	Rechnung wurde gedruckt und vom Einkäufer erfasst.
<i>Szenario</i>	<ol style="list-style-type: none"> <li>1 Einkäufer initiiert drucken der Einkaufsliste</li> <li>2 System wertet aus wie viele Zutaten aufgrund von Vorbestellungen notwendig sein werden. <i>Einkäufer vermerkt weitere Zutaten nach Absprache mit dem Koch auf der Einkaufsliste.</i></li> <li>3 Einkäufer verlässt das Restaurant mit Einkaufsliste.</li> </ol>

<i>Name</i>	erstelle Tagesbilanz
<i>Primäre Aktoren</i>	Buchhalter
<i>Vorbedingungen</i>	Buchhalter hat sich am System authentifiziert und ist eingeloggt. Eine Einkaufsabrechnung wurde vom Einkäufer erstellt und dem Buchhalter übermittelt. Alle Rechnungen von zubereiteten Speisen wurden dem Buchhalter übermittelt.
<i>Nachbedingungen</i>	
<i>Szenario</i>	<ol style="list-style-type: none"> <li>1 Buchhalter erfasst gekaufte Zutaten und deren Preise in einem Eingabeformular welches der Einkaufsabrechnung nachempfunden ist.</li> <li>2 Buchhalter erfasst benützte Zutaten, dabei werden die Speisen welche zubereitet wurden mit ihrer Anzahl und der Rechnungsnummer in einem Formular eingetragen.</li> <li>3 <i>Extension Point: drucke Fehlende Rechnungen</i> <i>Kellner wiederholt die Schritte 2-3 bis alle Zutaten abgearbeitet worden sind.</i></li> <li>4 Buchhalter erfasst verdorbene Zutaten</li> <li>5 System aktualisiert das Lagerjournal</li> <li>6 System rechnet Tagesbilanz aus und druck diese aus.</li> </ol>

## Anwendungsfalldiagramm

Das Anwendungsfalldiagramm beschreibt einen Ausschnitt der Funktionen des Systems und ist gemäß UML 2 spezifiziert. Jeder Anwendungsfall beschreibt Verhalten, das vom Software-System erwartet wird. Die grün markierten Anwendungsfälle sind in den Anwendungsfallbeschreibungen näher spezifiziert.

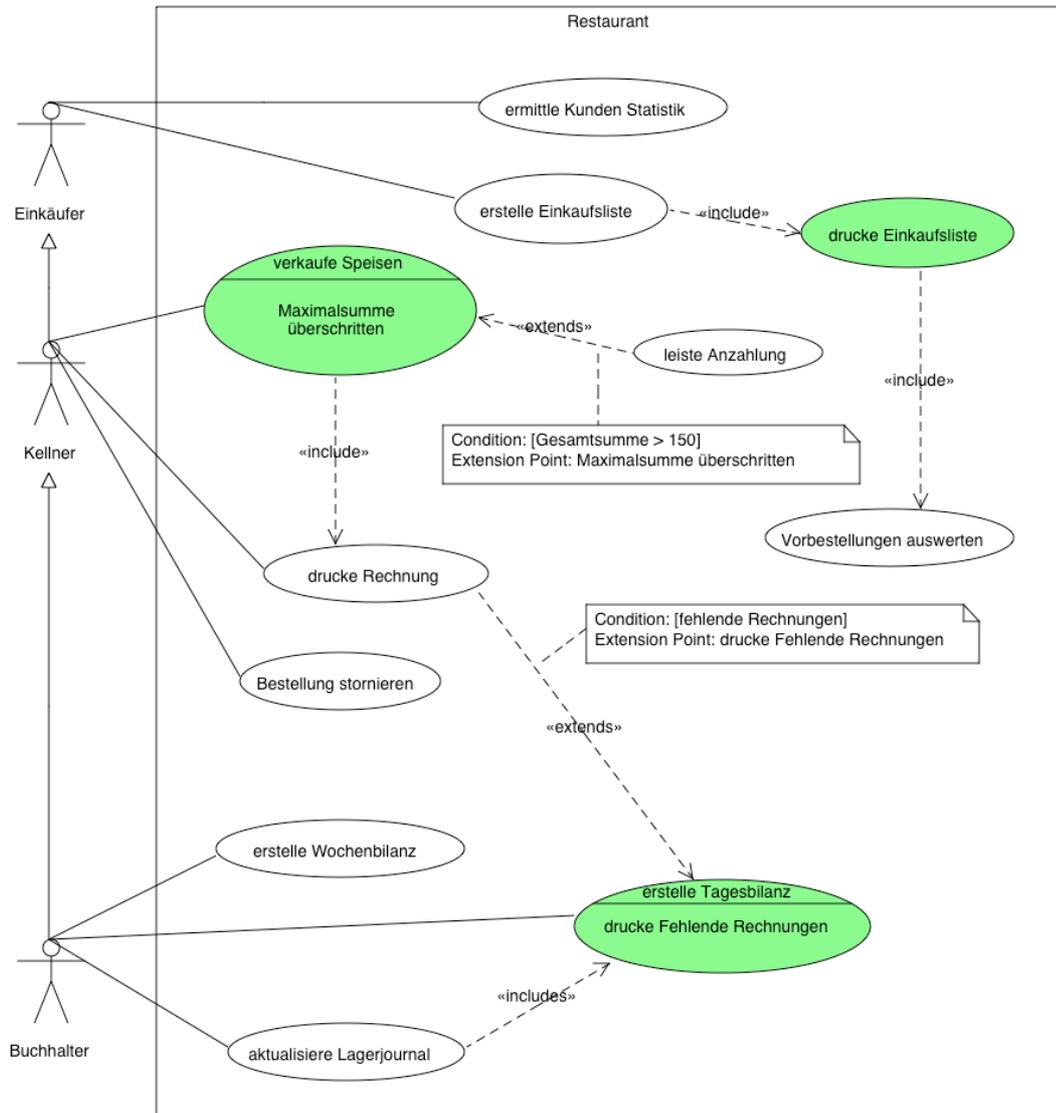


Abb. 1: Anwendungsfalldiagramm zum Beispiel „Restaurant“.

## Domänenmodell

Die Klassen sind gemäß UML 2 spezifiziert, sie bilden die anwendungs-spezifische Repräsentation der Informationen mit welchen das Softwaresystem arbeitet. Das Domänenmodell zeigt keine Softwareobjekte sondern eine Visualisierung von Konzepten in der Anwendungs-Domäne.

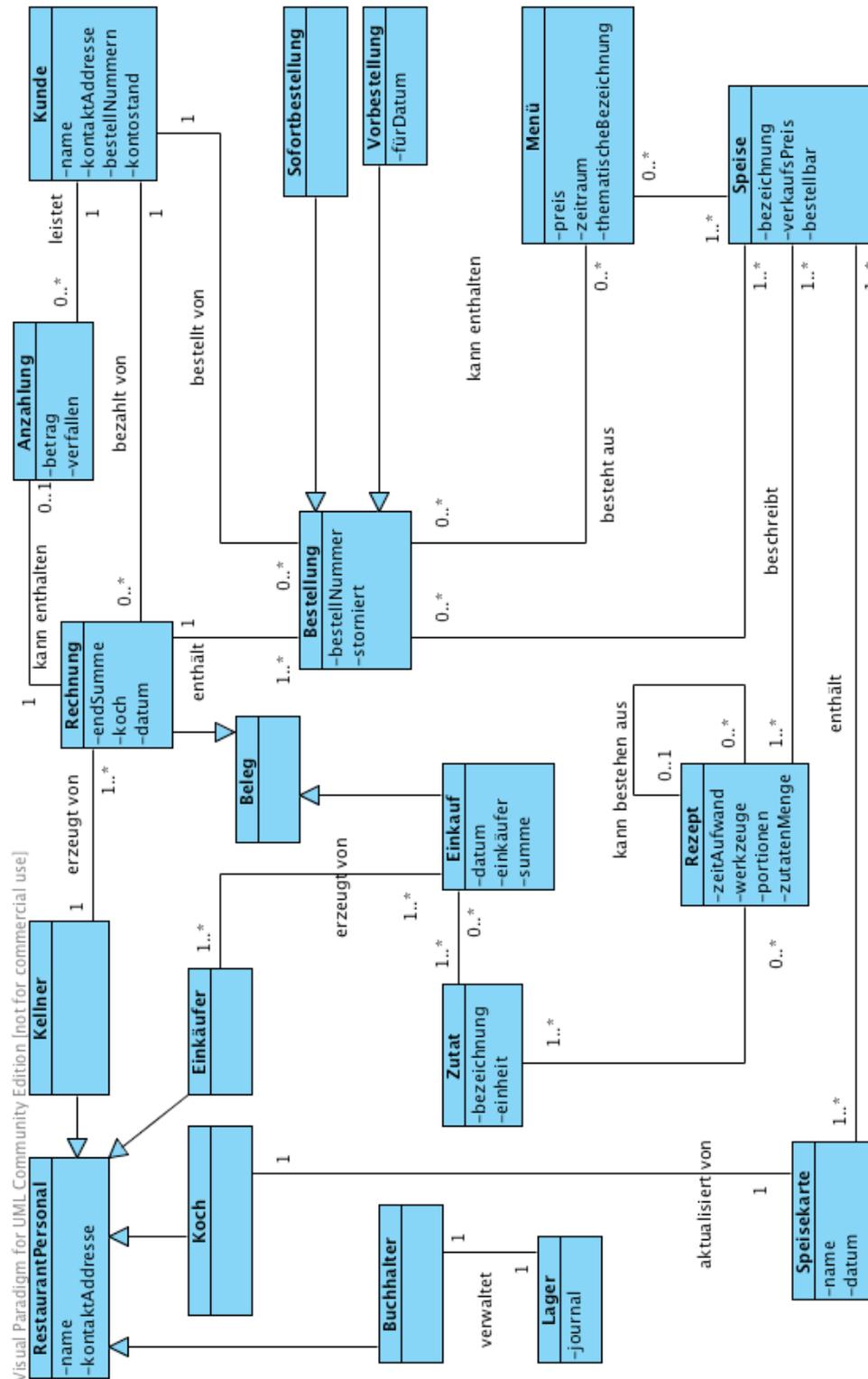


Abb. 2: Domänenmodell zum Beispiel „Restaurant“.

### Missionskritische Testfälle

Test Nr.	Typ	Kurzbeschreibung	Vorbedingungen	Eingabewerte / Parameter	Aktionen	Erwartete Ergebnisse
1	N F	Rechnung bezahlen für mehrere Bestellungen	Kunde und Bestellungen existieren	-Liste: Bestellung.id -Kunde.id -Kellner.id -Rechnung.bezahlt	-Bestellungen auswählen. -Option „bezahlen“ auswählen. -Bezahlung entgegennehmen. -Abrechnungsvorgang beenden, Rechnung als bezahlt markieren.	Rechnung ist persistent gespeichert, mit dem Kunden und den Bestellungen verbunden und als beglichen markiert. Metainformationen der Bestellungen wie etwa Koch und Zeit/Datum sind auch gespeichert. Rechnung erhält einen generierten Primärschlüssel von der DB.
2	N F	Mehrere Rechnungen bezahlen	Kunde und Bestellungen existieren	-Liste: Bestellung.id -Kunde.id -Kellner.id -Liste: Rechnung.bezahlt	-Bestellungen auswählen. -Option „getrennt bezahlen“ auswählen. -Bezahlung entgegennehmen. -Abrechnungsvorgang beenden, mehrere Rechnungen müssen als bezahlt markiert werden.	Für jede Bestellung ist eine Rechnung in der DB angelegt. Metainformationen der Bestellungen wie etwa Koch und Zeit/Datum sind auch für jede Rechnung gespeichert. Alle Rechnungen erhalten einen generierten Primärschlüssel von der DB.
3	N F	Bestellung von Speise durchführen	Kunde existiert, Speise mit preis<250 existiert	-Speise.id -Bestellung.take_away -Bestellung.fürDatum	-Neue Bestellung wird erzeugt, es wird angegeben wann die Bestellung konsumiert wird und ob sie mitgenommen wird. -Eine Speise wird der Bestellung zugeordnet. -Bestellung wird gespeichert.	Bestellung ist in der Datenbank gespeichert und einem Kellner zugeordnet. Das Attribut Bestellung.storniert ist auf den Booleanwert ‚false‘ gesetzt, der Primärschlüssel wird von der DB automatisch belegt.
4	S F	Bestellung mit Anzahlung	Menüs sind zum zuteilen vorhanden (bestellbar)	-Liste: Menu.id -Bestellung.take_away -Bestellung.fürDatum	-Neue Bestellung erzeugen, take_Away und fürDatum spezifizieren. -Menüs auswählen und der Bestellung zuordnen bis der Gesamtpreis der Bestellung mehr als 250.- Euro beträgt	Die Bestellung kann nicht gespeichert werden bis der Anzahlungsbetrag in der Bestellung gesetzt ist. Wird dieser auch gesetzt kann die Bestellung in der DB gespeichert werden.
5	N F	Bestellung stornieren	Bestellung existiert	-Bestellung.id	-Bestellung auswählen -Option „stornieren“ auswählen.	Attribut Bestellung.storniert wird auf den „true“ gesetzt. Bestellung ist jetzt nicht mehr veränderbar und kann auch keiner Rechnung zugeordnet werden.

6	N F	Einkauf erfassen	Es liegt eine Einkaufsrechnung vor wo Menge und Preis vermerkt sind.	-Zutat.bezeichnung -Zutat.menge -Zutat.einheitspreis	-Bezeichnung sowie Einheit der Zutaten sind im Regelwerk des Systems enthalten und werden aus Listen ausgewählt -Menge und der Einheitspreis der Zutaten werden manuell Eingegeben.	Der Gesamtpreis einzelnen Zutaten sowie des Gesamteinkaufs wird vom System berechnet. Die Einkaufsabrechnung sowie deren Zutaten werden (wiederherstellbar) in der DB abgelegt. Der Lagerstand in der DB wurde automatisch aktualisiert, das Lagerjournal wurde manuell aktualisiert.
7	N F	neues Menü erstellen	Speisen sind vorhanden	-Menu.preis -Menu.zeitraum -Menu.bezeichnung -Liste: Speise.id	-Preis, Zeitraum und Bezeichnung des Menüs werden in einem Formular eingetragen -Speisen werden aus einer Liste ausgewählt und anschließend gespeichert.	Das Menü kapselt ein oder mehrere Speisen, wird in der DB angelegt und kann bestellt werden.
8	N F	Einkaufsliste erstellen und drucken	Bestellungen mit .für_Datum in der Zukunft existieren	-Liste: Zutaten.id -Zutat.id	-Eine neue Einkaufsliste wird erzeugt. -Zutaten welche aufgrund der vorhandenen Vorbestellungen notwendig sind werden abgefragt und von System ausgewertet. -Liste von benötigten Zutaten wird der Einkaufsliste zugeordnet. -Eine weitere Zutat wird manuell der Einkaufsliste zugeordnet -Einkaufsliste wird gedruckt	Der Benutzer erhält eine ausgedruckte Einkaufsliste auf welcher die Zutaten von Vorbestellungen sowie manuell hinzugefügte Zutaten vermerkt sind. Außerdem ist die benötigte Menge für jede Zutat vermerkt, auch wenn diese 1 beträgt.
9	S F	Speisekarte bearbeiten	Speisekarte existiert. System wird per Touch-Screen mit Latex-Handschuhen bedient.	-Speisekarte.id -Speisekarte.datum -Liste: Speise.id -Menu.id	-Speisekarte auswählen. -Mehrere Speisen und ein Menü werden aus Listen ausgewählt. -Speisekarte wird gespeichert.	Die Speisekarte ist persistent in der DB abgelegt und das Datum der Speisekarte wurde automatisch aktualisiert.
10	S F	Nachbestellen	Bestellung mit mindestens einer Speise existiert	-Bestellung.id	Bestellung wählen und Option „nachbestellen“ auswählen.	Das zuletzt bestellte Produkt wird noch einmal bestellt. Der Preis der Bestellung wird um den Produktpreis erhöht.