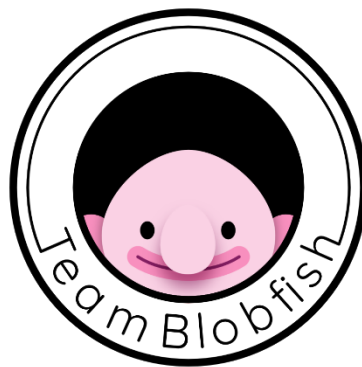


# Pflichtenheft V3.0

Von Team-Blobfish im Rahmen des SEP SS2021

22.06.2021

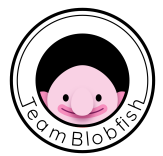


Kontakt: [HSTeamBlobfish@gmx.de](mailto:HSTeamBlobfish@gmx.de)

Kunde: Capgemini

Auftrag: Wartung und Weiterentwicklung der Beispiel-Applikation  
My-Thai-Star zum devonfw

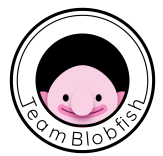
Autoren: Exoucia Mukubay, Florian Braasch, Maximilian Stinner



## Änderungshistorie

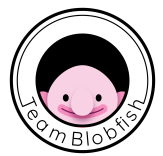
Version	Datum	Vorgenommene Änderungen	Autor
3.0	22.06.2021	Korrektur	Florian Braasch, Sven Reinemuth
2.5	22.06.2021	Ergänzung am Kapitel "Überlieferung" nach Kunden-Info am 22.06, Änderung von Anforderung "Passwort-Reset-Link versenden"	Maximilian Stinner
2.4	21.06.2021	"Weboberfläche" Vervollständigung	Maximilian Stinner
2.3	20.06.2021	kleinere Änderungen an Kapitel Produkteinsatz	Maximilian Stinner
2.2	18.06.2021	Architektur-Dokumente angepasst/aktualisiert und beschrieben	Maximilian Stinner
2.1	17.06.2021	direktes Feedback am 17.06. von Herr Schramm umgesetzt (Kapitel Produkteinsatz verlegen/anpassen, Kapitel Benutzeroberfläche hinzugefügt)	Maximilian Stinner
2.0	11.06.2021	weitere Änderungen anhand des Feedbacks, Anforderungen hinzugefügt nach Kundenabsprache, Korrektur	Maximilian Stinner, Florian Braasch, Exoucia Mukubay
1.3	10.06.2021	Feedback des Kunden und Managements einarbeiten	Maximilian Stinner, Florian Braasch
1.2	6.06.2021	erste Einarbeitung des Feedbacks	Maximilian Stinner
1.1	4.06.2021	Feedback als Kommentare eingearbeitet	Florian Braasch
1.0	12.05.2021	Korrektur	Sven Reinemuth, Maximilian Stinner, Florian Braasch, Exoucia Mukubay

Die erste Nachkommastelle der Versionsnummer wird jedes Mal wenn eine Änderung vollzogen wird hochgezählt. Ganzzahlige Versionsnummern stellen ein Inkrement dar, welches dem Kunden/Management abgegeben wird.

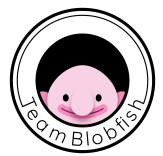


## Inhaltsverzeichnis

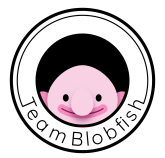
<b>1. Vorwort</b>	<b>7</b>
1.1 Anforderungsänderungen	7
<b>2. Einführung</b>	<b>8</b>
2.1 Firmenumfeld	8
2.2 devonfw und "My-Thai-Star"	8
<b>3. Ziel</b>	<b>9</b>
3.1 Zustand des "My-Thai-Star"-Systems	9
3.1.1 Anfangszustand	9
3.1.2 Ist-Zustand	9
3.2 Projekt Vision	9
3.3 Musskriterien	10
3.4 Abgrenzungskriterien	10
3.5 Wunschkriterien	10
<b>4. Produktleistungen</b>	<b>11</b>
4.1 Rollen und ihre Rechte	11
4.2 Use-Case Diagramm	12
4.3 Use-Case Beschreibungen	14
4.3.1 UC1 Bestellungen archivieren und auf das Bestellarchiv zugreifen	14
4.3.2 UC2 Bestellstatus editieren oder Bestellungen stornieren	15
4.3.3 UC3 Auf Administrationsoberfläche zugreifen	17
4.3.4 UC4 Benutzerkonten erstellen	18
4.3.5 UC5 Benutzerkonten löschen	20
4.3.6 UC6 Passwort-Reset-Link versenden	21
4.3.7 UC7 Passwörter setzen	22
4.3.8 UC8 Essen bestellen über Alexa-Inhouse	24
4.3.9 UC9 Speisekarte abfragen über Alexa-Inhouse	25
4.3.10 UC10 Servierzeiten festlegen über Alexa-Inhouse	27
4.3.11 UC11 Essen bestellen über Alexa-Home	29
4.3.12 UC12 Speisekarte abfragen über Alexa-Home	30
4.3.13 UC13 Tisch reservieren über Alexa-Home	30
4.3.14 UC14 Essen bestellen über Webanwendung	32



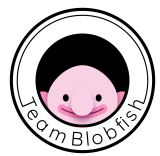
4.4 Zusammenfassung der Anforderungen mit Priorisierung	33
4.5 Anforderungen an das System	36
4.5.1 Funktionale Anforderungen an das System	36
4.5.1.1 Ansicht von archivierten Bestellungen	36
4.5.1.2 Archivierung von Bestellungen	37
4.5.1.3 Bestellstatus editieren	37
4.5.1.4 Bestellungen stornieren	38
4.5.1.5 Administrationsoberfläche	39
4.5.1.6 Benutzerkonten erstellen	39
4.5.1.7 Benutzerkonten löschen	40
4.5.1.8 Passwort-Reset-Link versenden	40
4.5.1.9 Passwörter setzen	41
4.5.1.10 Bestellungen erhalten Status	42
4.5.1.11 Benutzerrollen zuweisen	43
4.5.1.12 Essen bestellen über Alexa-Inhouse	43
4.5.1.13 Abfragen der Speisekarte über Alexa-Inhouse	44
4.5.1.14 Tisch reservieren über Alexa-Home	44
4.5.1.15 Essen bestellen über Alexa-Home	45
4.5.1.16 Abfrage der Speisekarte über Alexa-Home	45
4.5.1.17 Gast über die Webanwendung bei Reservierung einladen	46
4.5.1.18 Essen bestellen über Webanwendung	46
4.5.1.19 Die Buchungs-ID wird dem Kunden angezeigt	47
4.5.1.20 Servierzeit über Webanwendung bei Bestellung festlegen	47
4.5.1.21 Bugs beheben	48
4.5.1.22 Servierzeiten festlegen über Alexa-Inhouse	48
4.5.1.23 Servierzeiten festlegen über Alexa-Home	49
4.5.1.24 Namen und TableID anzeigen	49
4.5.1.25 Bezahlstatus editieren	50
4.5.1.26 Bestellungen filtern	50
4.5.1.27 Benutzerregistration	51
4.5.1.28 Daten der Benutzerkonten editieren	51
4.5.1.29 Die Buchungs-ID wird automatisch in die Bestellung eingetragen	52
4.5.1.30 Darstellung auf mobilen Interface	52



4.5.2 Nicht-Funktionale Anforderungen an das System	53
4.5.2.1 Kellner-Cockpit übersichtlich machen	53
4.5.2.2 Kellner-Cockpit leicht bedienbar machen	54
4.5.2.3 Administrationsoberfläche übersichtlich machen	54
4.5.2.4 Administrationsoberfläche leicht bedienbar machen	55
4.5.2.5 Zuverlässige Alexa-Inhouse Steuerung	55
4.5.2.6 Alexa-Inhouse Steuerung leicht verständlich machen	56
4.5.2.7 Alexa-Inhouse Steuerung intuitiv	56
4.5.2.8 Zuverlässige Alexa-Home Steuerung	57
4.5.2.9 Alexa-Home Steuerung leicht verständlich machen	57
4.5.2.10 Alexa-Home Steuerung intuitiv	58
4.5.2.11 Intuitiver Online Reservierungsvorgang	58
<b>5. Qualitätsanforderungen</b>	<b>59</b>
5.1 QR1 Codequalität	59
5.2 QR2 Vollständigkeit	59
5.3 QR3 Sprache	60
5.4 QR4 Modularität	60
5.5 QR5 Erweiterbarkeit	61
5.6 QR6 Usability	61
5.7 QR7 Ähnlichkeit	62
<b>6. Qualitätsziele</b>	<b>63</b>
<b>7. Produkteinsatz</b>	<b>64</b>
7.1 Weboberfläche	64
7.1.1 Waiter Cockpit	64
7.1.2 Administrationsoberfläche	66
7.1.3 Essen bestellen	67
7.1.4 Tisch reservieren	67
7.2 Kontextabgrenzung	68
7.3 Komponentendiagramm	69
7.4 Verteilungsdiagramm	70
<b>8. Auslieferung</b>	<b>71</b>
<b>9. Software Glossar</b>	<b>72</b>



<b>10. Glossar</b>	<b>72</b>
<b>11. Ergänzungen</b>	<b>75</b>
<b>12. Quellenverzeichnis</b>	<b>76</b>
<b>13. Abbildungsverzeichnis</b>	<b>76</b>
<b>14. Tabellenverzeichnis</b>	<b>76</b>
<b>15. Unterschriften</b>	<b>79</b>
15.1 Auftragnehmer	79
15.2 Auftraggeber	80



## 1. Vorwort

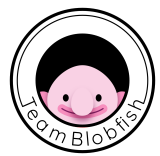
Das Pflichtenheft wird von der Gruppe Team Blobfish im Rahmen des Software-Entwicklungs-Projekt (SEP) an der Hochschule Mannheim bearbeitet. Im Pflichtenheft werden alle Verpflichtungen gegenüber dem Vertragspartner dokumentiert, dazu gehören unter anderem funktionale und nicht-funktionale Anforderungen und stellt die Vertragsgrundlage dar. Das Software-Entwicklungs-Projekt befasst sich mit der Wartung, Weiterentwicklung als auch Dokumentation des Beispielprojekts "My-Thai-Star", das mit der Nutzung des Devon-Frameworks entstanden ist.

### 1.1 Anforderungsänderungen

Da sich Änderungen im Verlauf eines Projektes durch geänderte Anforderungen oder Rahmenbedingungen ergeben können, ist es wichtig, eine Behandlung für diese zu dokumentieren.

Sollte es Änderungen seitens des Kunden geben, so sollten diese schnellstens dem Entwicklungsteam bekannt gegeben werden.

Nach Eingang der neuen Anforderung bespricht sich das Team intern über die Maßnahmen, die abhängig vom Umfang der Anforderungsänderung getroffen werden müssen. Muss-Anforderungen haben zum Beispiel eine höhere Auswirkung auf die Projektdurchführung als Kann-Anforderungen. Wichtig ist dann, dass die Priorität der neuen Anforderung vom Kunden definiert wird, damit das Team zusammen mit dem Kunden einschätzen kann, ob und welche Anforderungen stattdessen vernachlässigt werden können. Anforderungsänderungen werden bis zum 21.06.2021 entgegengenommen und werden nach gemeinsamer Einigkeit im Pflichtenheft (und/oder gegebenenfalls anderen Dokumenten) hinzugefügt.



## 2. Einführung

### 2.1 Firmenumfeld

Der Auftraggeber des Projekts ist Capgemini, das ein börsennotiertes Beratungs- und IT-Dienstleistungsunternehmen europäischen Ursprungs ist und den Sitz in Paris hat, jedoch in mehr als 50 Ländern tätig ist.

### 2.2 devonfw und “My-Thai-Star”

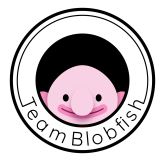
Die moderne Softwareentwicklung steht vor immer mehr Herausforderungen. Eine davon ist sicherlich Software schneller zu entwickeln. Durch die stetig wachsende Vielzahl an Technologien und den Anstieg der Komplexität wird viel Aufwand und Zeit in die Aufbereitung der Softwarearchitektur gesteckt.

Die von Capgemini initialisierte Open-Source Entwicklungsplattform devonfw soll den Softwareentwicklungsprozess standardisieren, indem viele Erfolgsmodelle miteinander verknüpft werden. Dadurch bekommt man ein Template, das die Einrichtung der Entwicklungsumgebung, mit allen wichtigen Technologien des Front- sowie Backends, übernimmt. Es müssen zusätzlich keine weitere Open-Source Tools und Entwicklungsumgebungen installiert werden, wodurch man als Entwickler direkt mit der Implementierung anfangen und schnelle Erfolge erzielen kann.

Das Framework wurde für das Restaurant-Management-System “My-Thai-Star” verwendet, welches ebenfalls ein von Capgemini initiiertes Open-Source Projekt darstellt. “My-Thai-Star” hat eine Weboberfläche, in der Kunden Tische reservieren, Freunde einladen und Essen bestellen können. Zudem gibt es weitere Rollen, den Kellner, den Administrator und den Manager, die sich in der Anwendung anmelden können und verschiedene Berechtigungen erhalten.

Weitere Infos zu den Rollen und deren Funktionen in der [Rollenübersicht](#) und den [Use-Cases](#).





## 3. Ziel

### 3.1 Zustand des “My-Thai-Star”-Systems

#### 3.1.1 Anfangszustand

Das “My-Thai-Star”-System wies zu Beginn des Projekts einen unfertigen Zustand auf.

Bekannte Bugs, die dem öffentlichen Git-Repository von “My-Thai-Star” zu entnehmen sind, haben Probleme verursacht.

Ein weiteres Problem stellte die fehlende Verfügbarkeit der Buchungs-ID dar, wodurch kein Essen vom Nutzer bestellt werden konnte.

Auch wurden einige Funktionalitäten für das Waiter-Cockpit, eine Ansicht der Administrator-Oberfläche mit den Funktionen, die der Berechtigung des Administrators zuteil werden, sowohl Funktionen, die der Nutzer haben sollte, noch nicht implementiert.

Ein Buchungs- oder Bestellvorgang konnte zudem nicht über Alexa-Skills von Amazon erfolgen, da die Funktion und ihre Anbindung ans “My-Thai-Star”-System noch nicht implementiert wurde.

#### 3.1.2 Ist-Zustand

Der Ist-Zustand beschreibt den aktuellen Zustand des “My-Thai-Star”-Systems, der zum Zeitpunkt der letzten Version des Pflichtenhefts besteht.

Zum aktuellen Zustand wurden alle Bugs außer der “Entity references across components #13”-Bug gelöst.

Zudem kann die Buchungs-ID vom Nutzer bei der Tischreservierung eingesehen werden.

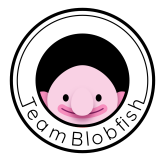
Alle übrigen im Pflichtenheft beschriebenen funktionalen Muss-Anforderungen zu dem Waiter-Cockpit, der Admin Oberfläche und Alexa wurden bereits erfüllt.

Der Fokus wird auf das Testen dieser Anforderungen und auf die Soll-/Kann-Anforderungen gelegt.

## 3.2 Projekt Vision

Das Ziel des Projekts ist es grundsätzlich einen Beitrag zur Entwicklung von “My-Thai-Star” zu leisten. Die Vision stellt ein Produkt dar, das seinen Nutzen in einem echten, modernisierten Restaurant finden könnte.

Neben den grundlegenden Funktionen eines Restaurant-Management-Systems, wird die Sprachsoftware und künstliche Intelligenz “Alexa” in das System integriert. Dadurch erhält der Anwender die Möglichkeit, Bestellungen per Sprachbefehl aufzunehmen oder sogar konkrete Fragen zu stellen, worauf “Alexa Skills” interaktive Antworten erzeugt.



### 3.3 Musskriterien

Muss Kriterien umfassen alle aufgeführten Muss-Anforderungen ohne die das Produkt in seinen Funktionen eingeschränkt ist. Das betrifft die im [Anfangszustand](#) benannten Probleme. Konkret kann man die Muss-Anforderungen aus dem Kapitel Funktionale Anforderungen an das System" (siehe Kapitel 4.5.1 Funktionale Anforderungen an das System) entnehmen.

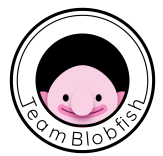
### 3.4 Abgrenzungskriterien

Die Entwicklung von "My-Thai-Star" wird um folgendes abgegrenzt:

- Es werden lediglich die geforderten Bugs gefixt und weitere Bugs werden nur dann betrachtet, wenn sie die Implementierung der geforderten Funktionalitäten stören
- Wir beziehen uns bei der Arbeit mit "My-Thai-Star" nur auf die von devonfw verwendeten Technologien
- Alexa muss nur auf Deutsch kommunizieren können
- Auf der Weboberfläche müssen alle Texte auf Englisch verfügbar sein. Bis auf Deutsch, dessen Implementierung eine Kann-Anforderung ist, werden keine weiteren Sprachen unterstützt.

### 3.5 Wunschkriterien

Die Wunschkriterien umfassen alle Kann-Anforderungen die in diesem Dokument erwähnt werden, sie gelten als optional und werden erfüllt, wenn dem Team Zeit dafür bleibt und alle Muss- und Soll-Kriterien bereits erfüllt sind.



## 4. Produktleistungen

In diesem Abschnitt werden sämtliche Leistungen der “My-Thai-Star“-Applikation aufgelistet und ausführlich spezifiziert.

### 4.1 Rollen und ihre Rechte

*Tabelle 1: Rollen und ihre Rechte/Funktionalitäten*

Rolle	Rechte und Funktionalitäten
Administrator	Zugriff auf Administrationsoberfläche
Kellner	Zugriff auf “Kellner Cockpit” inklusive Bestellsarchiv
Kunde	Reservierungs- und Bestellungsverlauf
Manager	Zugriff auf das Bestellsarchiv, Vorhersagemanagement und Clustermanagement

## 4.2 Use-Case Diagramm

Das Use-Case Diagramm stellt die wichtigsten Funktionalitäten der „My-Thai-Star“-Applikation dar.

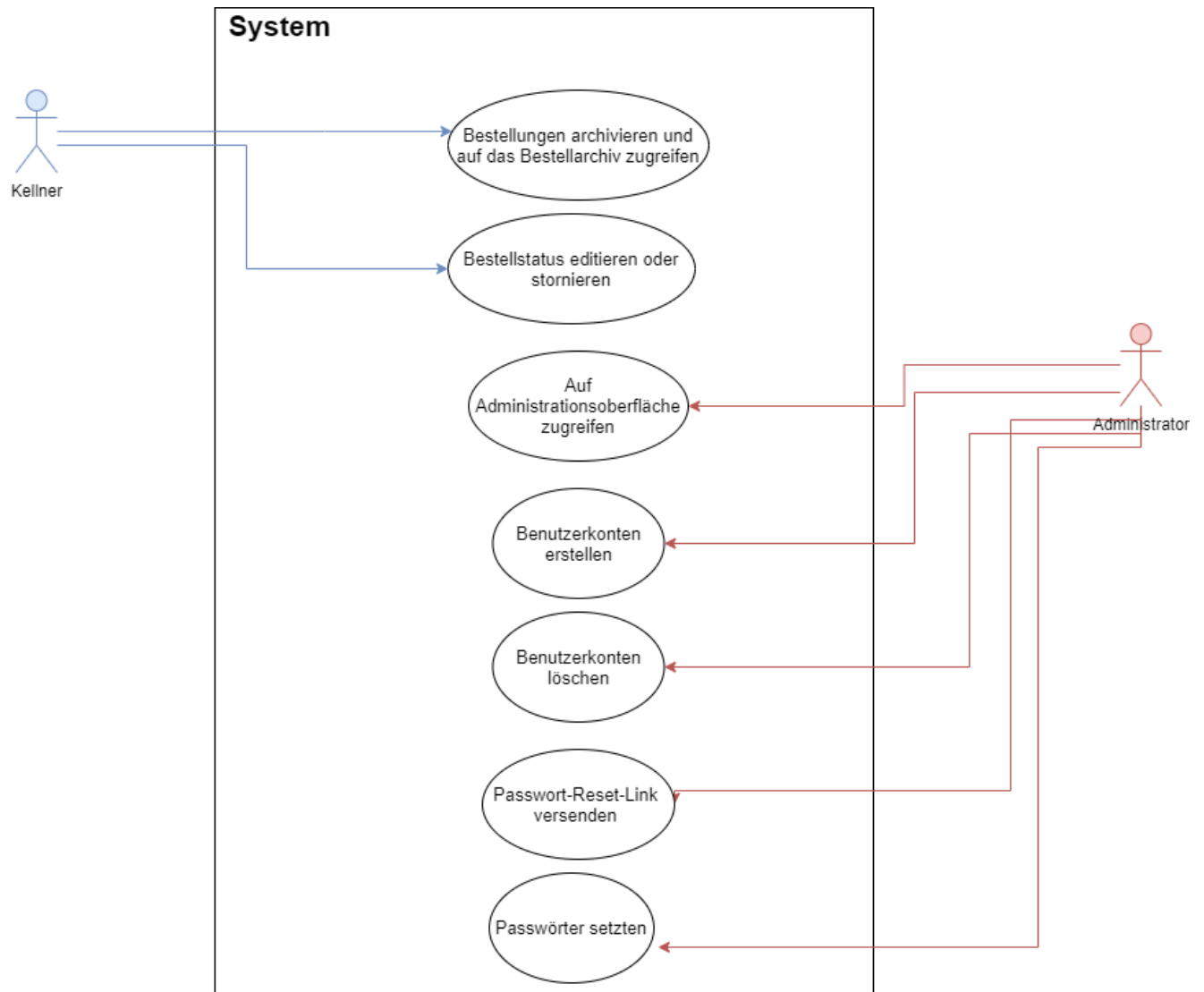


Abbildung 1: Use-Case Diagramm - Kellner und Administrator

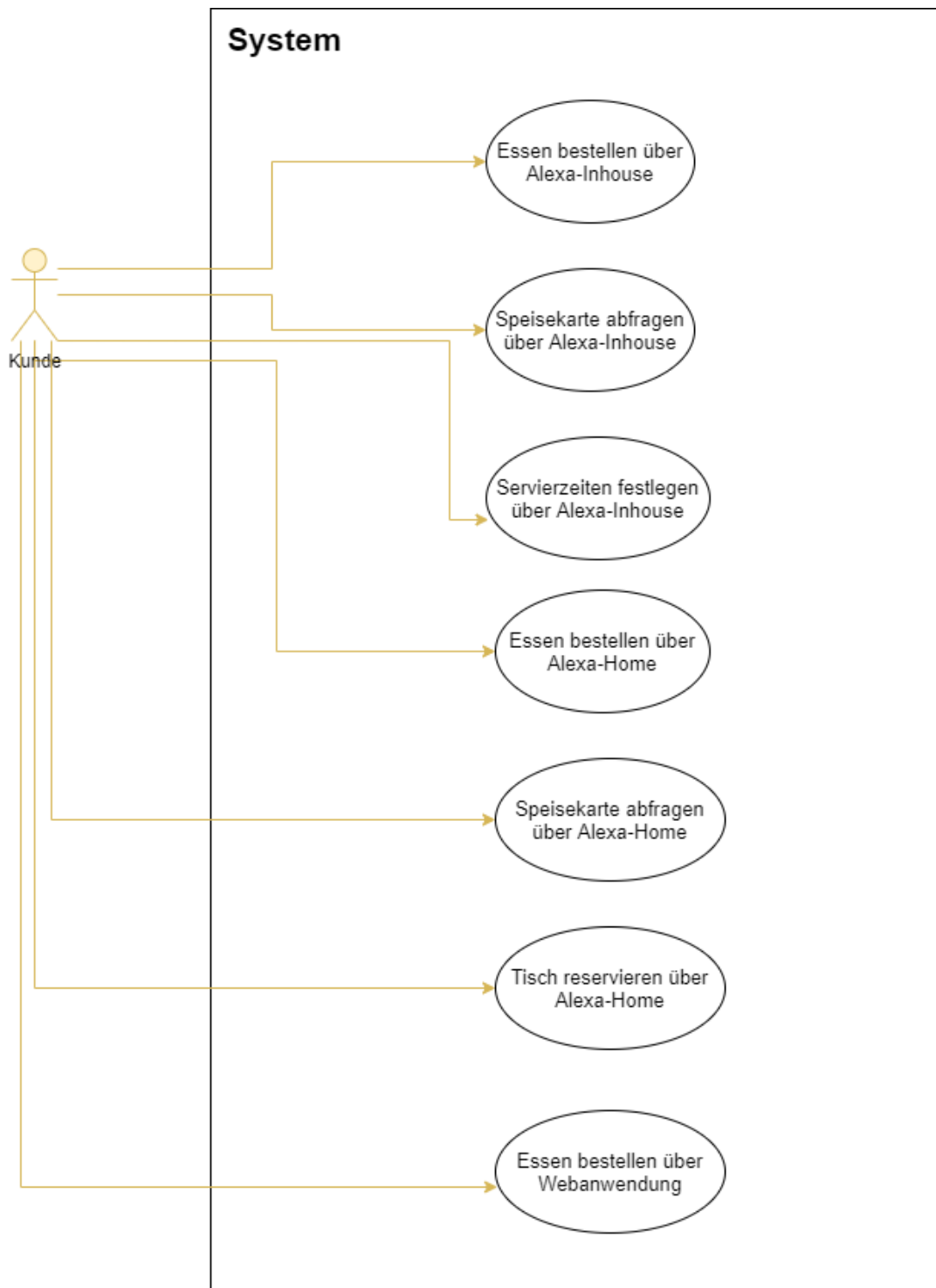
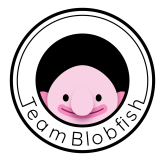


Abbildung 2: Use-Case Diagramm - Kunde

Die Rolle des Managers und dessen Funktionalitäten Vorhersage- und Clustermanagement werden nicht dargestellt, da diese schon vorher implementiert wurden und nicht zu den Anforderungen des Kunden Capgemini zählen.

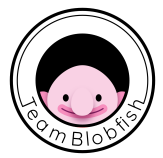


## 4.3 Use-Case Beschreibungen

### 4.3.1 UC1 Bestellungen archivieren und auf das Bestellarchiv zugreifen

Tabelle 2: Use Case Beschreibung – Bestellungen archivieren und auf das Bestellarchiv zugreifen

Abschnitt	Beschreibung
Bezeichner	Bestellungen archivieren und auf das Bestellarchiv zugreifen
Autoren	Exoucia Mukubay, Florian Braasch
Ziel	Es muss eine Ansicht mit archivierten Bestellungen für den Kellner sichtbar sein.
Akteure	Kellner
Vorbedingung	Es muss eine Funktion implementiert werden, sodass Bestellungen archiviert werden können.
Ergebnis	Der Kellner kann Bestellungen archivieren und alle archivierten Bestellungen ansehen.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Kellner meldet sich an ein Kellner-Benutzerkonto an.</li> <li>2. Das System überprüft die Logindaten, sind diese Korrekt wird der Benutzer angemeldet. Die Übersicht für den Kellner wird geladen. Das System baut die "Bestellungsübersicht" auf</li> <li>3. Der Kellner kann Bestellungsstatus und einen unabhängigen Bezahlstatus zuweisen.</li> <li>4. Wenn das System eine Bestellung einen als "Abgeschlossen" definierten Status erreicht, wird die Bestellung in das Bestellsarchiv übertragen und aus den "offenen" entfernt.</li> <li>5. Mit einem Klick auf den Menüpunkt "Bestellarchiv" kann der Kellner die Ansicht wechseln.</li> <li>6. Das System baut die "Archivansicht" auf.</li> </ol>
Nachbedingung	Eine archivierte Bestellung ist nicht mehr in der Bestellübersicht einsehbar, sondern nur noch im Bestellsarchiv
Ausnahme Abläufe	-
Regeln	Der Kellner muss eingeloggt sein.

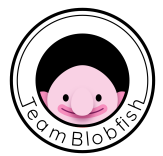


Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• FA 1</li> <li>• FA 2</li> <li>• FA 10</li> <li>• FA 26</li> </ul>
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• NFA 1</li> <li>• NFA 2</li> </ul>

#### 4.3.2 UC2 Bestellstatus editieren oder Bestellungen stornieren

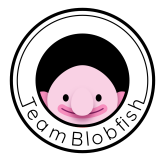
Tabelle 3: Use-Case Beschreibung – Bestellungen verwalten

Abschnitt	Beschreibung
Bezeichner	Bestellstatus editieren oder Bestellungen stornieren
Autoren	Exoucia Mukubay, Florian Braasch
Ziel	Der Kellner muss die Möglichkeit haben, "offene" Bestellungen zu editieren und stornieren.
Akteure	Kellner
Vorbedingung	Es müssen die Funktionen "Bestellung editieren" und "Bestellung stornieren" vorhanden sein.
Ergebnis	Der Kellner kann manuell den Status der offenen Bestellungen ändern oder die Bestellung stornieren.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Kellner meldet sich an ein Kellner-Benutzerkonto an.</li> <li>2. Das System überprüft die Logindaten, sind diese Korrekt wird der Benutzer angemeldet. Die Übersicht für den Kellner wird geladen. Das System baut die "Bestellungsübersicht" auf.</li> <li>3. Der Kellner hat eine Ansicht von allen Bestellungen vor sich und kann nun über Schnellauswahl Felder den Status der Bestellung ändern über einen Knopf auch Stornieren.</li> </ol>
Nachbedingung	Bestellstatus wurde geändert oder eine Bestellung wurde storniert. Ist der Bestellstatus auf "Abgeschlossen" geändert worden oder wurde eine Bestellung storniert wird diese in das Bestellungsarchiv aufgenommen.



Ausnahme Abläufe	<p>Bei der Bedienung macht der Benutzer einen Fehler:</p> <ol style="list-style-type: none"> <li>3. Der Kellner storniert oder ändert den Status der Falschen Bestellung.</li> <li>4. Die Falschen Änderungen werden in die Datenbank übernommen.</li> <li>5. Dem Kellner fallen die Fehler auf. Und kann solange er die Bestellansicht nicht verlassen hat die vorgenommenen Änderungen nacheinander, über eine eingefügte Schaltfläche zurücknehmen.</li> <li>6. Die zurückgenommenen Änderungen werden in die Datenbank übernommen.</li> </ol> <p>Oder:</p> <ol style="list-style-type: none"> <li>3. Der Kellner storniert oder ändert den Status der Falschen Bestellung.</li> <li>4. Die Falschen Änderungen werden in die Datenbank übernommen.</li> <li>5. Dem Kellner fallen die Fehler auf. Er hat jedoch die Bestellansicht zwischenzeitlich verlassen. Der Kellner wählt die betroffene Bestellung aus.</li> <li>6. Die Detailansicht der ausgewählten Bestellung wird aufgebaut.</li> <li>7. Der Kellner kann über ein Dropdown-Menü den entsprechend Richtigen Status auswählen. Darüber auch eine Falsche Stornierung zurücknehmen.</li> <li>8. Die zurückgenommenen Änderungen werden in die Datenbank übernommen.</li> </ol>
Regeln	Der Kellner muss eingeloggt sein.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>● FA 3</li> <li>● FA 4</li> <li>● FA 10</li> <li>● FA 24</li> <li>● FA 25</li> <li>● FA 26</li> </ul>
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"> <li>● NFA 1</li> <li>● NFA 2</li> </ul>

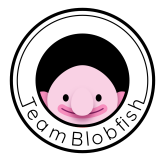




### 4.3.3 UC3 Auf Administrationsoberfläche zugreifen

Tabelle 4: Use-Case Beschreibung – Auf Administrationsoberfläche zugreifen

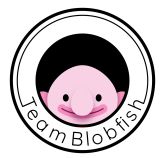
Abschnitt	Beschreibung
Bezeichner	Auf Administrationsoberfläche zugreifen
Autoren	Exoucia Mukubay, Florian Braasch
Ziel	Der Administrator möchte eine eigene Administrationsoberfläche haben, um seine Aufgaben zu erfüllen.
Akteure	Administrator
Vorbedingung	Die Administrationsoberfläche muss implementiert sein.
Ergebnis	Der Administrator hat eine eigene Administrationsoberfläche.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Administrator meldet sich mit seinem Benutzerkonto an.</li> <li>2. Das System erkennt anhand der Rollen, welche Rechte der Administrator hat. Die Administrationsoberfläche wird angezeigt.</li> </ol>
Ausnahme Abläufe	-
Regeln	Nur der Administrator kann auf die Administrationsoberfläche zugreifen.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• FA 5</li> </ul>
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• NFA 3</li> <li>• NFA 4</li> </ul>



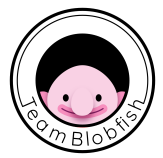
#### 4.3.4 UC4 Benutzerkonten erstellen

Tabelle 5: Use-Case Beschreibung – Benutzerkonten erstellen

Abschnitt	Beschreibung
Bezeichner	Benutzerkonten erstellen
Autoren	Exoucia Mukubay, Florian Braasch
Ziel	Der Administrator muss kann Benutzerkonten zu erstellen.
Akteure	Administrator
Vorbedingung	Die Funktionen „Benutzerkonten erstellen“, und „Benutzerrolle zuweisen“ müssen implementiert sein.
Ergebnis	Der Administrator kann Benutzerkonten erstellen.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Administrator meldet sich mit seinem Benutzerkonto an.</li> <li>2. Das System überprüft die Logindaten, sind diese korrekt wird der Benutzer angemeldet. Die Administrationsoberfläche hat eine Ansicht von allen registrierten Nutzerkonten.</li> <li>3. Der Administrator kann über eine Schaltfläche den Dialog zum Nutzer erstellen aufrufen und die Benutzerrolle zuweisen. Die entsprechenden Daten sowie die gewünschte Benutzerrolle werden eingetragen und an die Datenbank übermittelt.</li> <li>4. Das System legt den Nutzer in der Datenbank an und aktualisiert die Ansicht des Administrators.</li> </ol>
Nachbedingung	Das neu angelegte Benutzerkonto ist im System hinterlegt.
Ausnahmeabläufe	-
Regeln	Nur der Administrator kann Benutzerkonten erstellen und die Benutzerrolle zuweisen.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• FA 6</li> <li>• FA 11</li> <li>• FA 28</li> </ul>



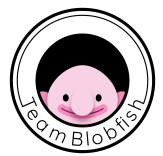
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"><li>• NFA 3</li><li>• NFA 4</li></ul>
---------------------------------	---



#### 4.3.5 UC5 Benutzerkonten löschen

Tabelle 6: Use-Case Beschreibung – Benutzerkonten löschen

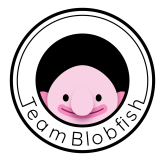
Abschnitt	Beschreibung
Bezeichner	Benutzerkonten löschen
Autoren	Exoucia Mukubay
Ziel	Der Administrator kann Benutzerkonten löschen.
Akteure	Administrator
Vorbedingung	Die Funktionen „Benutzerkonten löschen“.
Ergebnis	Der Administrator kann Benutzerkonten löschen.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Administrator meldet sich mit seinem Benutzerkonto an.</li> <li>2. Das System überprüft die Logindaten, sind diese Korrekt wird der Benutzer angemeldet. Der Administrationsoberfläche hat eine Ansicht von allen registrierten Benutz Konten.</li> <li>3. Der Administrator kann über eine Schaltfläche den Dialog zum Benutzer löschen aufrufen. Die entsprechenden Daten werden an die Datenbank übermittelt.</li> <li>4. Das System löscht den Benutzer in der Datenbank und aktualisiert die Ansicht des Administrators.</li> </ol>
Nachbedingung	Das gelöschte Benutzerkonto ist im System nicht mehr sichtbar.
Ausnahmeabläufe	-
Regeln	Nur der Administrator kann Benutzerkonten außer sein eigenes Konto löschen.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• FA 7</li> </ul>
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• NFA 3</li> <li>• NFA 4</li> </ul>



#### 4.3.6 UC6 Passwort-Reset-Link versenden

Tabelle 7: Use-Case Beschreibung – Passwort-Reset-Link versenden

Abschnitt	Beschreibung
Bezeichner	Passwort-Reset-Link versenden
Autoren	Exoucia Mukubay
Ziel	Der Administrator muss die Möglichkeit haben, Passwort-Reset-Links zu versenden.
Akteure	Administrator und Benutzer
Vorbedingung	Eine Administrationsoberfläche und die Funktion „Passwort-Reset-Link verschicken“ muss implementiert sein.
Ergebnis	Der Administrator kann Passwort-Reset-Links versenden, sodass der Nutzer die Möglichkeit bekommt sein Passwort zurückzusetzen.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Administrator meldet sich mit seinem Benutzerkonto an.</li> <li>2. Das System überprüft die Logindaten, sind diese Korrekt wird der Benutzer angemeldet. Der Administrationsoberfläche hat eine Ansicht von allen registrierten Nutzerkonten.</li> <li>3. Der Administrator kann über eine Schaltfläche den Dialog zum Nutzer, den Passwort-Reset-Link verschicken.</li> <li>6. Das System sendet dem Nutzer per E-Mail den Passwort-Reset Link.</li> <li>5. Der Nutzer klickt auf den Link und wird auf die Webseite des Systems weitergeleitet.</li> <li>6. Der Nutzer gibt sein neues Passwort ein.</li> <li>7. Die entsprechenden Daten werden an die Datenbank übermittelt.</li> <li>8. Das System aktualisiert das Passwort des Nutzers in der Datenbank.</li> </ol>
Nachbedingung	Das neue Passwort des Nutzers wird im System hinterlegt.
Ausnahmeabläufe	-
Regeln	Nur der Administrator kann den Passwort-Reset-Link versenden.

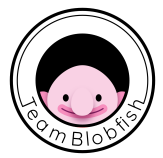


<b>Qualitäten</b>	Vollständig
<b>Funktionale Anforderungen</b>	<ul style="list-style-type: none"> <li>• FA 8</li> </ul>
<b>Nicht-Funktionale Anforderungen</b>	<ul style="list-style-type: none"> <li>• NFA 3</li> <li>• NFA 4</li> </ul>

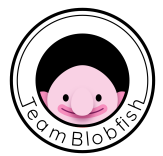
#### 4.3.7 UC7 Passwörter setzen

Tabelle 8: Use-Case Beschreibung – Passwörter setzen

Abschnitt	Beschreibung
Bezeichner	Passwörter setzen
Autoren	Exoucia Mukubay
Ziel	Der Administrator muss die Möglichkeit haben Passwörter von anderen Nutzern zu setzen.
Akteure	Administrator
Vorbedingung	Eine Administrationsoberfläche und die Funktion „Passwörter setzen“ müssen implementiert sein.
Ergebnis	Der Administrator kann Passwörter von anderen Nutzern setzen.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Administrator meldet sich mit seinem Benutzerkonto an.</li> <li>2. Das System überprüft die Logindaten, sind diese korrekt wird der Benutzer angemeldet. Der Administrationsoberfläche hat eine Ansicht von allen registrierten Nutzerkonten.</li> <li>3. Der Administrator kann über eine Schaltfläche den Dialog zum Nutzer, das Passwort setzen aufrufen. Das neue Passwort wird an die Datenbank übermittelt.</li> <li>4. Das System aktualisiert das Passworts des Nutzers in der Datenbank an und aktualisiert die Ansicht des Administrators, sodass das neue Passwort sichtbar ist.</li> </ol>
Nachbedingung	Der Benutzer kann sich mit dem gesetzten Passwort anmelden



Regeln	Nur der Administrator kann Passwörter setzen.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"><li>• FA 9</li></ul>
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"><li>• NFA 3</li><li>• NFA 4</li></ul>

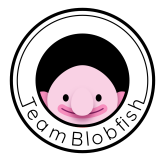


#### 4.3.8 UC8 Essen bestellen über Alexa-Inhouse

Tabelle 9: Use-Case Beschreibung – Essen bestellen über Alexa-Inhouse

Abschnitt	Beschreibung
Bezeichner	Essen bestellen über Alexa-Inhouse
Autoren	Exoucia Mukubay
Ziel	Alexa nimmt im Restaurant die Bestellung des Kunden auf.
Akteure	Kunde
Vorbedingung	Alexa Skills müssen auf dem Alexa-Echo Gerät installiert sein und gestartet werden.
Ergebnis	Der Kunde kann über Alexa Essen bestellen.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Kunde ruft den Alexa-Skill auf.</li> <li>2. Alexa beginnt die Interaktion und begrüßt den Nutzer.</li> <li>3. Der Kunde möchte Essen bestellen.</li> <li>4. Alexa fragt nach der Tischnummer</li> <li>5. Kunde gibt Tischnummer an</li> <li>6. Alexa schaut wer den Tisch um diese Uhrzeit reserviert hat und fragt den Kunde ob die Daten für den reservierten Tisch auf den Kunde zutreffen.</li> <li>7. Kunde bestätigt und nimmt anschließend die Bestellung auf</li> <li>8. Alexa fragt nach dem Bestellvorgang welche Servierzeit der Kunde wünscht</li> <li>9. Kunde gibt Servierzeit an und Alexa schickt die Bestellung ab</li> </ol>
Nachbedingung	Die Bestellung wurde aufgenommen und ist mit den gewünschten Daten im Kellner-Cockpit ersichtlich.
Regeln	Der Kunde muss in verständlicher Sprache mit Alexa kommunizieren.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• FA 12</li> <li>• FA 13</li> </ul>



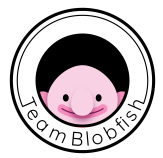


Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• NFA 5</li> <li>• NFA 6</li> <li>• NFA 7</li> </ul>
---------------------------------	---

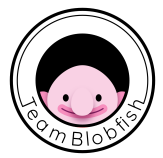
#### 4.3.9 UC9 Speisekarte abfragen über Alexa-Inhouse

Tabelle 10: Use-Case Beschreibung – Speisekarte abfragen über Alexa-Inhouse

Abschnitt	Beschreibung
Bezeichner	Speisekarte abfragen über Alexa-Inhouse
Autoren	Exoucia Mukubay
Ziel	Der Kunde kann Alexa die Speisekarte abfragen.
Akteure	Kunde
Vorbedingung	Alexa Skills müssen auf dem Alexa-Echo Gerät installiert sein und gestartet werden.
Ergebnis	Der Kunde fragt Alexa nach der Speisekarte.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Kunde sitzt an seinem gebuchten Platz und ruft den Alexa Skill auf</li> <li>2. Alexa beginnt die Interaktion und begrüßt den Nutzer.</li> <li>3. Der Kunde fragt nach der Speisekarte.</li> <li>4. Alexa lädt die gewünschten Daten aus der Datenbank und liest die Speisekarte vor.</li> </ol>
Nachbedingung	Nachdem die Speisekarte vorgelesen wurde, wird die Bestellung des Kunden aufgenommen (siehe oben <a href="#">UC9 "Essen Bestellen über Alexa-Home"</a> ).
Ausnahmeabläufe	-
Regeln	Der Kunde muss in verständlicher Sprache mit Alexa kommunizieren.
Qualitäten	Vollständig



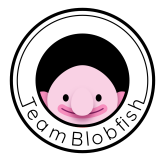
Funktionale Anforderungen	<ul style="list-style-type: none"><li>• FA 13</li></ul>
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"><li>• NFA 5</li><li>• NFA 6</li><li>• NFA 7</li></ul>



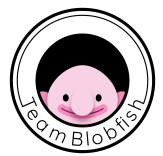
#### 4.3.10 UC10 Servierzeiten festlegen über Alexa-Inhouse

Tabelle 11: Use-Case Beschreibung – Servierzeiten festlegen über Alexa-Inhouse

Abschnitt	Beschreibung
Bezeichner	Servierzeiten festlegen über Alexa-Inhouse
Autoren	Exoucia Mukubay
Ziel	Der Kunde kann die Servierzeit(en) über Alexa festlegen.
Akteure	Kunde
Vorbedingung	Alexa Skills müssen auf dem Alexa-Echo Gerät installiert werden. Alexa hat die Bestellung des Kunden aufgenommen (siehe oben <a href="#">UC7 5.3.7 "Essen bestellen über Alexa-Inhouse"</a> ).
Nachbedingung	-
Ergebnis	Der Kunde über Alexa die Servierzeit(en) festlegen, wann gegessen wird.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Kunde legt die Servierzeit(en) fest.</li> <li>2. Alexa überprüft die Servierzeit mit der Buchungszeit und der aktuellen Zeit.</li> <li>3. Wenn die Servierzeit innerhalb der Buchungszeit +- 30min. und 30 min. hinter der aktuellen Zeit liegt, dann wird der Bestellvorgang abgeschlossen</li> </ol>
Nachbedingung	Die Bestellung wurde aufgenommen und ist mit den gewünschten Daten im Kellner-Cockpit ersichtlich.
Ausnahmeabläufe	-
Regeln	Der Kunde muss in verständlicher Sprache mit Alexa kommunizieren.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• FA 12</li> <li>• FA 22</li> </ul>



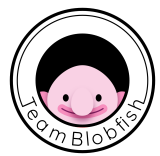
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"><li>• NFA 5</li><li>• NFA 6</li><li>• NFA 7</li></ul>
---------------------------------	---



#### 4.3.11 UC11 Essen bestellen über Alexa-Home

Tabelle 12: Use-Case Beschreibung - Essen bestellen über Alexa-Home

Abschnitt	Beschreibung
Bezeichner	Essen bestellen über Alexa-Home
Autoren	Exoucia Mukubay
Ziel	Der Kunde kann über Alexa-Home Essen bestellen.
Akteure	Kunde
Vorbedingung	Alexa Skills müssen auf dem Alexa-Echo Gerät installiert sein und gestartet werden. Bevor das Essen bestellt werden kann, muss zunächst ein Tisch gebucht werden (siehe <a href="#">unten UC12 "Tisch reservieren über Alexa-Home"</a> ).
Ergebnis	Der Kunde bestellt über Alexa-Home.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Kunde ruft den Alexa-Skill auf.</li> <li>2. Alexa beginnt die Interaktion und begrüßt den Nutzer.</li> <li>3. Der Kunde möchte Essen bestellen.</li> <li>4. Kunde hat die Auswahl sich das Essen liefern zu lassen oder sich das Essen für einen reservierten Tisch im Restaurant zu bestellen.</li> <li>5. Kunde kann mit Essen oder Trinken beginnen</li> <li>6. Kunde nimmt anschließend die Bestellung auf.</li> <li>7. Alexa fragt nach dem Bestellvorgang welche Servierzeit der Kunde wünscht</li> <li>8. Nach dem die Servierzeit angegeben wurde, wird die Bestellung abgeschickt</li> </ol>
Nachbedingung	Die Bestellung wurde aufgenommen und ist mit den gewünschten Daten im Kellner-Cockpit ersichtlich.
Regeln	Ein Tisch muss zunächst gebucht werden bevor eine Bestellung aufgenommen werden kann. Der Kunde muss in verständlicher Sprache mit Alexa kommunizieren.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• FA 15</li> <li>• FA 23</li> </ul>



Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• NFA 8</li> <li>• NFA 9</li> <li>• NFA 10</li> </ul>
---------------------------------	--

#### 4.3.12 UC12 Speisekarte abfragen über Alexa-Home

Tabelle 13: Use-Case Beschreibung – Speisekarte abfragen über Alexa-Home

Abschnitt	Beschreibung
Bezeichner	Speisekarte abfragen über Alexa-Home
Autoren	Exoucia Mukubay
Ziel	Der Kunde kann Alexa die Speisekarte abfragen.
Akteure	Kunde
Vorbedingung	Alexa Skills müssen auf dem Alexa-Echo Gerät installiert sein und gestartet werden.
Ergebnis	Der Kunde fragt Alexa nach der Speisekarte.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Kunde ruft den Alexa-Skill auf.</li> <li>2. Alexa beginnt die Interaktion und begrüßt den Nutzer.</li> <li>3. Der Kunde fragt nach der Speisekarte.</li> <li>4. Alexa lädt die gewünschten Daten aus der Datenbank und liest die Speisekarte vor.</li> </ol>
Nachbedingung	Nachdem die Speisekarte vorgelesen wurde, wird die Bestellung des Kunden aufgenommen (siehe oben <a href="#">UC10 "Essen bestellen über Alexa-Home"</a> ).
Ausnahmeabläufe	-
Regeln	Der Kunde muss in verständlicher Sprache mit Alexa kommunizieren.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• FA 16</li> </ul>
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• NFA 8</li> <li>• NFA 9</li> <li>• NFA 10</li> </ul>

#### 4.3.13 UC13 Tisch reservieren über Alexa-Home

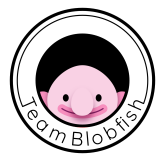
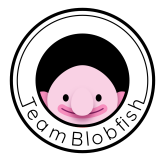


Tabelle 14: Use-Case Beschreibung – Tisch reservieren über Alexa-Home

Abschnitt	Beschreibung
Bezeichner	Tisch reservieren über Alexa-Home
Autoren	Exoucia Mukubay
Ziel	Der Kunde kann einen Tisch reservieren über Alexa-Home.
Akteure	Kunde
Vorbedingung	Alexa Skills müssen auf dem Alexa-Echo Gerät installiert sein und gestartet werden.
Ergebnis	Ein Tisch wird über Alexa-Home reserviert.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Kunde ruft den Alexa Skill auf. <ol style="list-style-type: none"> <li>1. Alexa beginnt die Interaktion und begrüßt den Nutzer.</li> </ol> </li> <li>2. Der Kunde möchte einen Tisch reservieren. <ol style="list-style-type: none"> <li>3. Alexa fragt nach Datum, Uhrzeit und Anzahl der Gäste.</li> </ol> </li> <li>4. Der Kunde beantwortet die Fragen. <ol style="list-style-type: none"> <li>5. Alexa nimmt die Reservierung auf, prüft die aufgenommenen Daten und fragt den Kunden, ob diese korrekt sind.</li> </ol> </li> <li>6. Der Kunde bestätigt die Daten. <ol style="list-style-type: none"> <li>7. Alexa leitet die Daten an die Datenbank weiter.</li> </ol> </li> </ol>
Nachbedingung	Die Reservierung wurde aufgenommen und ist mit den gewünschten Daten im Kellner-Cockpit ersichtlich.
Ausnahmeabläufe	-
Regeln	Die Voraussetzung um eine Bestellung aufzunehmen, muss zunächst ein Tisch reserviert werden. Der Kunde muss in verständlicher Sprache mit Alexa kommunizieren.
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>● FA 14</li> </ul>
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"> <li>● NFA 8</li> <li>● NFA 9</li> <li>● NFA 10</li> </ul>

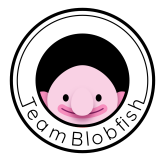


#### 4.3.14 UC14 Essen bestellen über Webanwendung

Tabelle 15: Use-Case Beschreibung – Essen bestellen über Webanwendung

Abschnitt	Beschreibung
Bezeichner	Essen bestellen über Webanwendung
Autoren	Exoucia Mukubay
Ziel	Es muss möglich sein, dass der Kunde über die Webanwendung Essen bestellen kann.
Akteure	Kunde
Vorbedingung	Die Funktion „Essen bestellen“ wurde schon implementiert.
Ergebnis	Der Kunde kann über die Webanwendung Essen bestellen.
Hauptablauf	<ol style="list-style-type: none"> <li>1. Der Kunde besucht die Webseite und bestellt ein Gericht es besteht die Möglichkeit Freunde einzuladen.</li> <li>2. Das My Thai Star-System erhält die Bestellung. Im Kellner-Cockpit wird die Bestellung aufgeführt.</li> </ol>
Nachbedingungen	Die Bestellung wird im Backend dem Bestellverzeichnis hinzugefügt.
Regeln	-
Qualitäten	Vollständig
Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• FA 18</li> <li>• FA 19</li> <li>• FA 20</li> <li>• FA 29</li> </ul>
Nicht-Funktionale Anforderungen	<ul style="list-style-type: none"> <li>• NFA 11</li> </ul>





#### 4.4 Zusammenfassung der Anforderungen mit Priorisierung

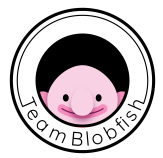
Im Folgenden werden alle funktionalen und nicht-funktionalen Anforderungen, die wir in Kann-, Soll- und Muss-Kriterien unterteilt haben, aufgelistet.

Das Muss-Kriterium sagt aus, dass die Anforderung vom System umgesetzt werden muss. Eine Anforderung im Soll-Kriterium, sollte, wenn es zeitlich möglich ist, umgesetzt werden. Das Kann-Kriterium beschreibt einen Zusatz, dass umgesetzt werden kann, wenn ein großer Teil der Muss- und Soll-Anforderungen bereits erfüllt wurden.

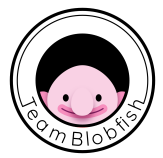
Zusätzlich gibt es eine Spalte „Priorität“, mit Hilfe derer die Anforderungen nach Wichtigkeit sortierbar sind. Muss-Kriterien sind je mit Prioritäten ab 1 eingeteilt, Soll-Kriterien mit Prioritäten 2-3 und Kann-Kriterien mit Prioritäten 3-4. Die Priorität 1 ist die dabei höchste und sinkt mit steigender Zahl. An der Priorisierung ist zu erkennen, dass der Schwerpunkt der Anforderungen Umsetzung vermehrt auf den einzelnen Komponenten liegt.

*Tabelle 16: Zusammenfassung der funktionalen Anforderungen mit Prioritäten*

Nr.	Typ	Beschreibung	Priorität	Kann- / Soll- / Muss- Kriterium
1	Funktional	Ansicht von archivierten Bestellungen	1	Muss
2	Funktional	Archivierung von Bestellungen	1	Muss
3	Funktional	Bestellungen editieren	1	Muss
4	Funktional	Bestellungen stornieren	1	Muss
5	Funktional	Administrationsoberfläche	1	Muss
6	Funktional	Benutzerkonten erstellen	1	Muss
7	Funktional	Benutzerkonten löschen	1	Muss
8	Funktional	Passwort-Reset-Link versenden	1	Muss
9	Funktional	Passwörter setzen	1	Muss
10	Funktional	Bestellungen erhalten Status	1	Muss
11	Funktional	Benutzerrollen zuweisen	1	Muss



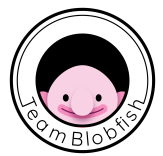
12	Funktional	Essen bestellen über Alexa-Inhouse	1	Muss
13	Funktional	Abfragen der Speisekarte über Alexa-Inhouse	1	Muss
14	Funktional	Tisch reservieren über Alexa-Home	1	Muss
15	Funktional	Essen bestellen über Alexa-Home	1	Muss
16	Funktional	Abfrage der Speisekarte über Alexa-Home	1	Muss
17	Funktional	Gast über die Webanwendung bei Reservierung einladen	1	Muss
18	Funktional	Essen bestellen über Webanwendung	1	Muss
19	Funktional	Die Buchungs-ID wird dem Kunden angezeigt	1	Muss
20	Funktional	Servierzeit über Webanwendung bei Bestellung festlegen	1	Muss
21	Funktional	Bugs beheben	1	Muss
22	Funktional	Servierzeiten festlegen über Alexa-Inhouse	2	Soll
23	Funktional	Servierzeiten festlegen über Alexa-Home	2	Soll
24	Funktional	Namen und TableID anzeigen	2	Soll
25	Funktional	Bezahlstatus editieren	2	Soll
26	Funktional	Bestellungen filtern	2	Soll
27	Funktional	Benutzerregistration	3	Soll
28	Funktional	Daten der Nutzerkonten editieren	3	Kann
29	Funktional	Die Buchungs-ID wird automatisch in die Bestellung eingetragen	3	Kann



30	Funktional	Darstellung auf mobilem Interface	4	Kann
----	------------	-----------------------------------	---	------

Tabelle 17: Zusammenfassung der nicht-funktionalen Anforderungen mit Prioritäten

Nr.	Typ	Beschreibung	Priorität	Kann- / Soll- / Muss- Kriterium
1	Nicht-Funktional	Kellner-Cockpit übersichtlich machen	2	Soll
2	Nicht-Funktional	Kellner-Cockpit leicht bedienbar machen	3	Soll
3	Nicht-Funktional	Administrationsoberfläche übersichtlich machen	2	Soll
4	Nicht-Funktional	Administrationsoberfläche leicht bedienbar machen	3	Soll
5	Nicht-Funktional	Zuverlässige Alexa-Inhouse Steuerung	2	Soll
6	Nicht-Funktional	Alexa-Inhouse Steuerung leicht verständlich machen	3	Soll
7	Nicht-Funktional	Alexa-Inhouse Steuerung intuitiv	2	Soll
8	Nicht-Funktional	Zuverlässige Alexa-Home Steuerung	2	Soll
9	Nicht-Funktional	Alexa-Home Steuerung leicht verständlich machen	3	Soll
10	Nicht-Funktional	Alexa-Home Steuerung intuitiv	2	Soll
11	Nicht-Funktional	Intuitiver Onlinereservier Vorgang	2	Soll



## 4.5 Anforderungen an das System

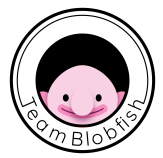
Die hier dargestellten funktionalen (FA) und nicht-funktionalen Anforderungen (NFA) stellen sämtliche Funktionalitäten des Gesamtsystems „My-Thai-Star“ dar.

### 4.5.1 Funktionale Anforderungen an das System

#### 4.5.1.1 Ansicht von archivierten Bestellungen

Tabelle 18: FA – Ansicht von archivierten Bestellungen

Nummer:	1
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Kellner hat eine Ansicht mit archivierten Bestellungen.
Rational:	Der Kellner braucht für die Abrechnung eine Bestellungsübersicht von allen Bestellungen, auch die, die archiviert wurden.
Use Case:	UC1
Fit Kriterium:	Der Kellner kann eine Ansicht mit archivierten Bestellungen öffnen und damit arbeiten.



#### 4.5.1.2 Archivierung von Bestellungen

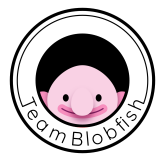
Tabelle 19: FA – Archivierung von Bestellungen

Nummer:	2
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Eine Bestellung ist archiviert, wenn der Bestellstatus auf „Abgeschlossen“ oder „Abgebrochen“ stehen. Das Archiv ist definiert als eine Sammlung von Bestellungen, die einen Endstatus erreicht haben.
Rational:	Der Kellner muss die Möglichkeit haben eine Bestellung archivieren zu können.
Use Case:	UC1
Fit Kriterium:	Die Bestellungen werden ins Bestellungs-Archiv aufgenommen wenn diese einen o.g. Endstatus erreichen.

#### 4.5.1.3 Bestellstatus editieren

Tabelle 20: FA – Bestellstatus editieren

Nummer:	3
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Kellner kann „offene“ Bestellungen editieren, den Status ändern.
Rational:	Falls der Kunde etwas bestellt hat und im Nachhinein noch nachträglich an seiner Bestellung was ändern möchte, kann der Kellner diese Bestellung editieren.
Use Case:	UC2

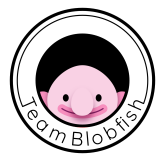


Fit Kriterium:	Der Kellner ist in der Lage in der Bestellübersicht den Status einer Bestellung ändern.
----------------	---

#### 4.5.1.4 Bestellungen stornieren

Tabelle 21: FA – Bestellungen stornieren

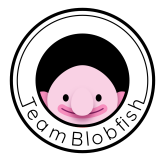
Nummer:	4
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Kellner kann „offene“ Bestellungen stornieren, Bestellungen aufheben
Rational:	Der Kunde bestellt versehentlich ein Gericht was er nicht wollte. Somit muss der Kellner die fehlerhafte Bestellung stornieren.
Use Case:	UC2
Fit Kriterium:	Der Kellner ist in der Lage Bestellungen zu stornieren, bzw. diese abzubereiten.



#### 4.5.1.5 Administrationsoberfläche

Tabelle 22: FA - Administrationsoberfläche

Nummer:	5
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Administrator möchte eine eigene Administrationsoberfläche haben, um seine Aufgaben zu erfüllen.
Rational:	Der Administrator braucht eine eigene Oberfläche, um Nutzerkonten und Bestellungen zu verwalten.
Use Case:	UC3
Fit Kriterium:	Der Administrator hat Zugriff auf die Administrationsoberfläche in der dieser Benutzerkonten verwalten kann.



#### 4.5.1.6 Benutzerkonten erstellen

Tabelle 23: FA – Benutzerkonten erstellen

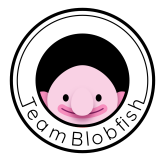
Nummer:	6
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Administrator muss die Möglichkeit haben Benutzerkonten zu erstellen.
Rational:	Der Administrator muss Nutzerkonten für die Mitarbeiter anlegen.
Use Case:	UC4
Fit Kriterium:	Der Administrator ist in der Lage neue Benutzerkonten anzulegen.

#### 4.5.1.7 Benutzerkonten löschen

Tabelle 24: FA – Benutzerkonten löschen

Nummer:	7
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Administrator muss die Möglichkeit haben Benutzerkonten zu löschen. Davon ausgenommen ist das eigene Adminkonto.
Rational:	Benutzerkonten, die nicht gebraucht werden, sollten gelöscht werden.
Use Case:	UC5
Fit Kriterium:	Der Administrator ist in der Lage Benutzerkonten zu löschen.





#### 4.5.1.8 Passwort-Reset-Link versenden

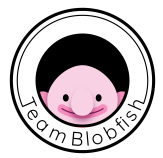
Tabelle 25: FA – Passwort-Reset-Link versenden

Nummer:	8
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Administrator kann Passwort-Reset-Links versenden.
Rational:	Falls der Nutzer sein Passwort nicht mehr weiß, sollte ein Link zu gesendet werden können, um den Nutzer die Möglichkeit anzubieten, ein neues Passwort zu setzen.
Use Case:	UC6
Fit Kriterium:	Der Administrator ist in der Lage den Benutzern ein Passwort-Reset-Link zu senden.

#### 4.5.1.9 Passwörter setzen

Tabelle 26: FA – Passwörter setzen

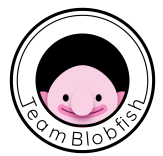
Nummer:	9
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Administrator kann Passwörter setzen
Rational:	-
Use Case:	UC7
Fit Kriterium:	Der Administrator ist in der Lage Passwörter für Benutzer konnten setzen.



#### 4.5.1.10 Bestellungen erhalten Status

*Tabelle 27: FA - Bestellungen erhalten Status*

Nummer:	10
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Bestellungen müssen einen Status bekommen.

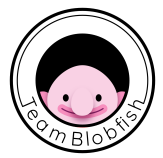


Rational:	<p>Bestellung müssen einen Bestellstatus und einen Zahlungsstatus haben.</p> <p>Ein Bestellstatus besteht aus den folgenden Begriffen: „Aufgenommen“, „In Zubereitung“, „Bereit“, „Wird ausgeliefert“, „Essen ist ausgeliefert“, „Abgeschlossen“ und „Abgebrochen“.</p> <p>Ein Bezahlstatus besteht aus den folgenden Begriffen: „Ausstehenden“, „Bezahlt“ und „Erstattet“.</p>
Use Case:	UC 1, UC2
Fit Kriterium	Jeder Bestellung werden ein Bestellstatus und ein Bezahlstatus gegeben. Es werden immer je einer vergeben.

#### 4.5.1.11 Benutzerrollen zuweisen

Tabelle 28: FA – Benutzerrollen zuweisen

Nummer:	11
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Administrator muss die Möglichkeit haben jedem neuen Benutzer eine Rolle zuweisen zu können.



Rational:	Jede Rolle hat seine eigene Funktionalität und Rechte, die nicht jeder Benutzer haben sollte. Es gibt folgende Rollen: Kellner, Admin, Kunde und Manager.
Use Case:	UC4
Fit Kriterium:	Der Administrator ist in der Lage bei erstellen eines Benutzerkontos diesem eine Rolle zuweisen.

#### 4.5.1.12 Essen bestellen über Alexa-Inhouse

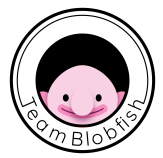
Tabelle 29: FA – Essen bestellen über Alexa-Inhouse

Nummer:	12
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Es muss möglich sein, dass Kunden über Alexa Essen bestellen können.
Rational:	Im Restaurant steht auf jedem Tisch eine Alexa, die Bestellungen aufnehmen soll.
Use Case:	UC8, UC10
Fit Kriterium:	Es ist möglich über Alexa ein Essen zu bestellen.

#### 4.5.1.13 Abfragen der Speisekarte über Alexa-Inhouse

Tabelle 30: FA – Abfragen der Speisekarte über Alexa-Inhouse

Nummer:	13
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Es muss über Alexa möglich sein, dass eine Abfrage der Speisekarte gestartet wird.



Rational:	Im Restaurant steht auf jedem Tisch eine Alexa, die Bestellungen aufnehmen soll. Da keine Speisekarten vorliegen, muss der Kunde Alexa nach der Speisekarte abfragen.
Use Case:	UC8, UC9
Fit Kriterium:	Ein Nutzer ist in der Lage über Alexa die Speisekarte des Restaurants abfragen.

#### 4.5.1.14 Tisch reservieren über Alexa-Home

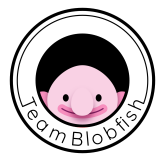
Tabelle 31: FA – Tisch reservieren über Alexa-Home

Nummer:	14
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Es muss möglich sein einen Tisch über Alexa reservieren zu können.
Rational:	Der Kunde möchte über Alexa einen Tisch reservieren statt über die Webanwendung.
Use Case:	UC13
Fit Kriterium:	Eine Reservierung über den Alexa Skill ist im Restaurant System korrekt eingegangen.

#### 4.5.1.15 Essen bestellen über Alexa-Home

Tabelle 32: FA – Essen bestellen über Alexa-Home

Nummer:	15
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Der Kunde kann über Alexa Essen bestellen.

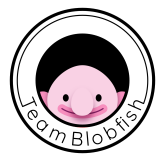


Rational:	-
Use Case:	UC11
Fit Kriterium:	Eine Bestellung über Alexa wird in der Bestellungen Ansicht angezeigt.

#### 4.5.1.16 Abfrage der Speisekarte über Alexa-Home

Tabelle 33: FA – Abfrage der Speisekarte über Alexa-Home

Nummer:	16
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Es muss über Alexa möglich sein, dass eine Abfrage der Speisekarte zu starten.
Rational:	Der Kunde hat die Speisekarte nicht vor sich und muss Alexa nach der Speisekarte abfragen.
Use Case:	UC12
Fit Kriterium:	Alexa kann nach Aufforderung die Speisekarte des Restaurants wiedergeben.



#### 4.5.1.17 Gast über die Webanwendung bei Reservierung einladen

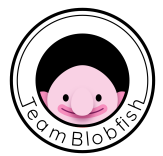
Tabelle 34: FA – Gast über die Webanwendung bei Reservierung einladen

Nummer:	17
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Es muss möglich sein, dass Kunden über die Webanwendung einen Gast oder mehrere Gäste einladen können
Rational:	-
Use Case:	UC14
Fit Kriterium:	Ein Kunde kann bei der Reservierung einen Gast/Freund einladen.

#### 4.5.1.18 Essen bestellen über Webanwendung

Tabelle 35: FA – Essen bestellen über Webanwendung

Nummer:	18
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Es muss möglich sein, dass der Kunde über die Webanwendung Essen bestellen kann.
Rational:	-
Use Case:	UC14
Fit Kriterium	Bestellungen über die Webanwendung werden an die Bestellungsübersicht des Restaurants übermittelt.



#### 4.5.1.19 Die Buchungs-ID wird dem Kunden angezeigt

Tabelle 36: FA – Die Buchungs-ID wird dem Kunden angezeigt

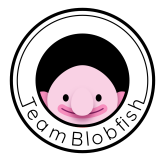
Nummer:	19
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Die Buchungs-ID, welche bei der Tischreservierung generiert wird, wird dem Kunden nach der Tischreservierung angezeigt.
Rational:	Für das Abschließen einer Bestellung benötigt der Kunde eine Buchungs-ID, die er mit der Bestellung abschicken muss.
Use Case:	UC 14
Fit Kriterium:	Der Kunde sieht die Buchungs-ID nach Abschluss einer Tischreservierung

#### 4.5.1.20 Servierzeit über Webanwendung bei Bestellung festlegen

Tabelle 37: FA - Servierzeit über Webanwendung bei Bestellung festlegen

Nummer:	20
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	Es muss möglich sein, dass Kunden, über die Webanwendung, Servierzeiten eine Bestellung hinzuzufügen.
Rational:	Der Kunde möchte dem Kellner mitteilen zu welchem Zeitpunkt er essen möchte.
Use Case:	UC14
Fit Kriterium:	Die Funktionalität, um Servierzeiten festzulegen können, müssen implementiert werden.





#### 4.5.1.21 Bugs beheben

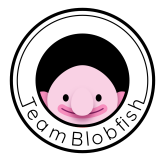
Tabelle 38: FA – Bugs beheben

Nummer:	21
Typ:	Funktionale Anforderung
Kriterium:	Muss-Kriterium
Beschreibung:	<p>folgende Bugs müssen behoben werden:</p> <ul style="list-style-type: none"> <li>- Entity references across components #13 (Der Bug wird nicht behoben, näheres <a href="#">hier</a>)</li> <li>- Object type vs. primitive types #15</li> <li>- Declaration of fields without types #19</li> <li>- Information disclosure, password plaintext storage #55</li> </ul>
Rational:	Aufgrund der Bugs kann es zu unerwünschten Fehlverhalten in der Applikation kommen
Use Case:	-
Fit Kriterium:	Bugs wurden behoben

#### 4.5.1.22 Servierzeiten festlegen über Alexa-Inhouse

Tabelle 39: FA – Servierzeiten festlegen über Alexa-Inhouse

Nummer:	22
Typ:	Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Es soll möglich sein, dass Kunden bei einer Essensbestellung eine Servierzeit festlegen können
Rational:	Der Kunde möchte nach seiner Ankunft einen Zeitpunkt festlegen wann er essen möchte.
Use Case:	UC10
Fit Kriterium:	Es ist möglich das ein Kunde bei einer Bestellung eine Servierzeit festlegt



#### 4.5.1.23 Servierzeiten festlegen über Alexa-Home

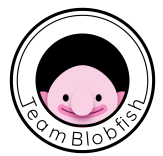
Tabelle 40: FA – Servierzeiten festlegen über Alexa-Home

Nummer:	23
Typ:	Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Es soll möglich sein, dass der Kunde bei seiner Essensbestellung eine Servierzeit festlegen kann.
Rational:	Der Kunde möchte dem Kellner mitteilen zu welchem Zeitpunkt er essen möchte.
Use Case:	UC11
Fit Kriterium:	Der Kunde kann dem Restaurant über den Alexa Skill eine Servierzeit mitgeben.

#### 4.5.1.24 Namen und TableID anzeigen

Tabelle 41: FA – Namen und TableID anzeigen

Nummer:	24
Typ:	Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Die Spalten "Name" und "TableID" werden in den Auflistungen der Tabellenansicht aller aktuellen Bestellungen und der Ansicht von archivierten Bestellungen angezeigt. Die "TableID" ersetzt die schon vorher implementierte Spalte "Email".
Rational:	Name und die TableID (Tischnummer) sind wichtige Informationen für den Kellner
Use Case:	UC2
Fit Kriterium:	Die Spalten "Name" und "TableID" werden angezeigt



#### 4.5.1.25 Bezahlstatus editieren

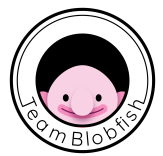
Tabelle 42: FA – Bezahlstatus editieren

Nummer:	25
Typ:	Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Der Kellner soll den Bezahlstatus bearbeiten können.
Rational:	Der Kellner muss von jedem Kunden wissen, wie der Bezahlstatus ist, damit es zu keinen Konflikten kommt.
Use Case:	UC2
Fit Kriterium:	Der Kellner kann den Bezahlstatus verändern.

#### 4.5.1.26 Bestellungen filtern

Tabelle 43: FA – Bestellung filtern

Nummer:	26
Typ:	Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Der Kellner soll in der Lage sein nach Name, TableID, Bestellstatus und Bezahlstatus zu filtern. Je nach Filterauswahl sollen die zutreffenden Bestellungen in der Auflistung angezeigt werden. Dies gilt für die Ansicht aller aktuellen Bestellungen und der Ansicht von archivierten Bestellungen.
Rational:	Der Kellner möchte nicht lange nach Bestellungen suchen. Dabei helfen Filteroptionen.
Use Case:	UC1, UC2
Fit Kriterium:	Nur die den Filteroptionen (Name, TableID, Bestellstatus, Bezahlstatus) zutreffenden Bestellungen werden angezeigt.



#### 4.5.1.27 Benutzerregistration

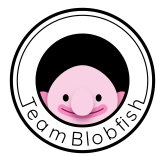
Tabelle 44: FA – Benutzerregistration

Nummer:	27
Typ:	Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Es soll möglich sein, dass Kunden ein Benutzerkonto registrieren können.
Rational:	Der Kunde will ein Benutzerkonto haben, um ein eigenes Profil mit Daten zu speichern.
Use Case:	-
Fit Kriterium:	Ein Kunde kann sich auf der "My-Thai-Star"-Website registrieren und auf dem erstellten Benutzerkonto anmelden. Ein Kunde kann sich bei der Registration keine Rechte geben.

#### 4.5.1.28 Daten der Benutzerkonten editieren

Tabelle 45: FA – Daten der Benutzerkonten editieren

Nummer:	28
Typ:	Funktionale Anforderung
Kriterium:	Kann-Kriterium
Beschreibung:	Der Administrator soll die Daten vorhandener Nutzerkonten ändern können. Dazu gehören der Name, die Email und die Rolle des Kontos.
Rational:	Da der Administrator vollen Zugriff auf alle Nutzerkonten hat, soll es möglich sein die Nutzerdaten zu editieren.
Use Case:	UC4
Fit Kriterium:	Der Administrator kann Nutzerdaten verändern.



#### 4.5.1.29 Die Buchungs-ID wird automatisch in die Bestellung eingetragen

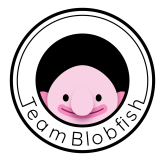
Tabelle 46: FA – Die Buchung-ID wird automatisch in die Bestellung eingetragen

Nummer:	29
Typ:	Funktionale Anforderung
Kriterium:	Kann-Kriterium
Beschreibung:	Die Buchungs-ID wird vom Kunden für das Abschließen einer Bestellung benötigt. Die Buchungs-ID wird bei einer Bestellung automatisch mitgesendet.
Rational:	Der Kunde will sich nicht mehr Arbeit machen und die Buchungs-ID manuell eingeben
Use Case:	-
Fit Kriterium:	Die Bestellung wird abgeschlossen und die Buchungs-ID erscheint im Backend unter der richtigen Bestellung.

#### 4.5.1.30 Darstellung auf mobilen Interface

Tabelle 47: FA – Darstellung auf mobilen Interface

Nummer:	30
Typ:	Funktionale Anforderung
Kriterium:	Kann-Kriterium
Beschreibung:	Die Webanwendung kann auf kleinen Bildschirmen (Mobile Geräte) problemlos verwendet werden
Rational:	Der Kunde möchte über die App auf seinem Tablet, Smartphone usw. Essen bestellen.
Use Case:	-
Fit Kriterium:	Elne Dynamische Web Umgebung des Produktes ist auch auf handelsüblichen Mobilen Endgeräten verwendbar.

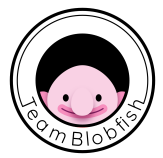


## 4.5.2 Nicht-Funktionale Anforderungen an das System

### 4.5.2.1 Kellner-Cockpit übersichtlich machen

Tabelle 48: NFA – Kellner-Cockpit übersichtlich machen

Nummer:	1
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Das Kellner-Cockpit soll übersichtlich aufgebaut sein.
Rational:	Der Kellner findet sich schnell zurecht.
Use Case:	UC1
Fit Kriterium:	Die Übersichtlichkeit des Kellner-Cockpit, wird mit Usability Tests ermittelt und gemessen. Das Kellner-Cockpit hat die Usability Test bestanden.



#### 4.5.2.2 Kellner-Cockpit leicht bedienbar machen

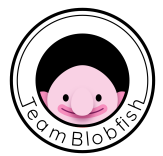
Tabelle 49: NFA – Kellner-Cockpit leicht bedienbar machen

Nummer:	2
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Das Kellner-Cockpit soll leicht zu bedienen sein.
Rational:	Der Kellner soll keine Schwierigkeiten beim Bedienen des Kellner-Cockpits haben.
Use Case:	UC1
Fit Kriterium:	Die Bedienbarkeit des Kellner-Cockpit, wird mit Usability Test ermittelt und gemessen. Das Kellner-Cockpit hat die Usability Test bestanden.

#### 4.5.2.3 Administrationsoberfläche übersichtlich machen

Tabelle 50: NFA – Administrationsoberfläche übersichtlich machen

Nummer:	3
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Die Administrationsoberfläche muss übersichtlich aufgebaut sein
Rational:	Der Administrator braucht einen klaren Überblick über alle Bestellungen und Nutzerkonten.
Use Case:	UC3
Fit Kriterium:	Die Usability für die Adminoberfläche werden erfolgreich abgeschlossen.



#### 4.5.2.4 Administrationsoberfläche leicht bedienbar machen

Tabelle 51: NFA – Administrationsoberfläche leicht bedienbar machen

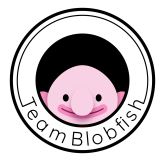
Nummer:	4
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Die Administrationsoberfläche soll leicht zu bedienen sein.
Rational:	Der Administrator soll keine Schwierigkeiten beim Bedienen der Oberfläche haben.
Use Case:	UC3
Fit Kriterium:	Die Usability Tests der Adminoberfläche werden erfolgreich abgeschlossen.

#### 4.5.2.5 Zuverlässige Alexa-Inhouse Steuerung

Tabelle 52: NFA – Zuverlässige Alexa-Inhouse Steuerung

Nummer:	5
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Alexa Steuerung soll zuverlässig funktionieren.
Rational:	Es darf nicht zu technischen Fehlern kommen.
Use Case:	UC7, UC8, UC9
Fit Kriterium:	Die Usability Tests und manuellen Tests der Alexa Skills wurden erfolgreich abgeschlossen.





#### 4.5.2.6 Alexa-Inhouse Steuerung leicht verständlich machen

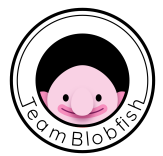
Tabelle 53: NFA – Alexa-inhouse Steuerung leicht verständlich machen

Nummer:	6
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Alexa Steuerung soll leicht zu verstehen sein.
Rational:	Es darf nicht zu Missverständnissen kommen.
Use Case:	UC7, UC8, UC9
Fit Kriterium:	Alexa Sprachausgaben bestehen die Usability Tests.

#### 4.5.2.7 Alexa-Inhouse Steuerung intuitiv

Tabelle 54: NFA – Alexa-Inhouse Steuerung intuitiv

Nummer:	7
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Alexa Steuerung soll intuitiv sein.
Rational:	Alexa soll intuitiv handeln.
Use Case:	UC7, UC8, UC9
Fit Kriterium:	Der Alexa Skill besteht die Usability Tests.



#### 4.5.2.8 Zuverlässige Alexa-Home Steuerung

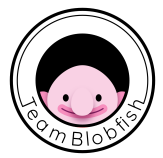
Tabelle 55: NFA – Zuverlässige Alexa-Home Steuerung

Nummer:	8
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Alexa-Home Steuerung soll zuverlässig funktionieren.
Rational:	Es darf nicht zu technischen Fehlern kommen.
Use Case:	UC10
Fit Kriterium:	Der Alexa-Home Skill besteht die Usability Tests.

#### 4.5.2.9 Alexa-Home Steuerung leicht verständlich machen

Tabelle 56: NFA – Alexa-Home Steuerung leicht verständlich machen

Nummer:	9
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Alexa-Home Steuerung soll leicht zu verstehen sein.
Rational:	Es darf nicht zu Missverständnissen kommen.
Use Case:	UC10
Fit Kriterium:	Der Alexa-Home Skill besteht die Usability Tests.



#### 4.5.2.10 Alexa-Home Steuerung intuitiv

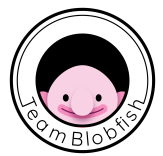
Tabelle 57: NFA – Alexa-Home Steuerung intuitiv

Nummer:	10
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Alexa-Home Steuerung soll intuitiv sen.
Rational:	Alexa-Home soll intuitiv handeln.
Use Case:	UC10
Fit Kriterium:	Der Alexa-Home Skill besteht die Usability Tests.

#### 4.5.2.11 Intuitiver Online Reservierungsvorgang

Tabelle 58: NFA – Intuitiver Online Reservierungsvorgang

Nummer:	11
Typ:	Nicht-Funktionale Anforderung
Kriterium:	Soll-Kriterium
Beschreibung:	Der Online Reservierung Vorgang soll intuitiv sein.
Rational:	-
Use Case:	UC11
Fit Kriterium:	Die Usability Tests der Webseite werden erfolgreich abgeschlossen.



## 5. Qualitätsanforderungen

### 5.1 QR1 Codequalität

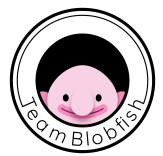
Tabelle 59: Qualitätsanforderung 1- Codequalität

Name	Codequalität
<b>Beschreibung</b>	Der Code des Systems soll wartbar, gut strukturiert und neue Implementationen ausreichend kommentiert sein
<b>Begründung</b>	Das System soll als Beispielprojekt für die devonfw IDE Umgebung dienen - zukünftig auch weiterentwickelt werden können - Codequalität soll Einarbeitungszeit verringern
<b>Erfüllungskriterium</b>	Alle Variablen besitzen selbsterklärende Namen, neue Implementation sind nach JavaDoc Standards zu dokumentieren. Der Code ist übersichtlich strukturiert und mit Tests versehen

### 5.2 QR2 Vollständigkeit

Tabelle 60: Qualitätsanforderung 2 - Vollständigkeit

Name	Vollständigkeit
<b>Beschreibung</b>	Das Gesamtsystem soll alle in Muss-Kriterien definierten funktionalen Anforderungen unterstützen
<b>Begründung</b>	Das Projekt "My Thai Star" macht nur Sinn und funktioniert als Ganzes, wenn sämtliche Muss-Anforderungen vorhanden sind
<b>Erfüllungskriterium</b>	Das System gilt als vollständig, wenn alle in diesem Dokument genannten Muss-Anforderungen erfüllt sind



### 5.3 QR3 Sprache

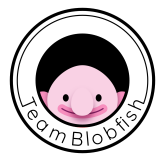
Tabelle 61: Qualitätsanforderung 3 - Sprache

Name	Sprache
<b>Beschreibung</b>	Das gesamte Projekt soll in Englisch programmiert und kommentiert werden
<b>Begründung</b>	Unser Kunde möchte ein verständliches Projekt. Es soll keine Vermischung zweier Sprachen geben
<b>Erfüllungskriterium</b>	Der komplette Code und die Kommentare sind in Englisch

### 5.4 QR4 Modularität

Tabelle 62: Qualitätsanforderung 4 - Modularität

QR4	Modularität
<b>Beschreibung</b>	Komponenten die durch unser Team hinzugefügt werden sollen dem modularen Aufbau der Applikation folgen
<b>Begründung</b>	Das Übergebende Projekt ist bereits modular aufgebaut und diese modularer Aufbau eine Stärke des devon-Framework ist. Weiterhin bringt die Modularität den Vorteil, dass Änderungen im Code keine Auswirkungen auf weitere Module haben.
<b>Erfüllungskriterium</b>	Neue Komponenten besitzen klare Schnittstellen und müssen nicht nachträglich auf hinzugefügte Komponenten verändert werden



## 5.5 QR5 Erweiterbarkeit

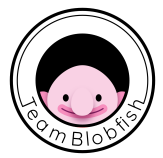
Tabelle 63: Qualitätsanforderung 5 - Erweiterbarkeit

QR5	Erweiterbarkeit
<b>Beschreibung</b>	Das System wird auch nach unserem Projekt erweiterbar sein
<b>Begründung</b>	Das System "My-Thai-Star" wird Aufgrund dessen das es Open-Source ist voraussichtlich in Zukunft weiterhin wachsen. Um dieses Wachstum nicht einzuschränken sollten neue Komponenten weiterhin problemlos erweitert werden können
<b>Erfüllungskriterium</b>	Neue Komponenten, dürfen nicht auf Grundlage veränderter Schnittstellen realisiert werden

## 5.6 QR6 Usability

Tabelle 64: Qualitätsanforderung 6 - Usability

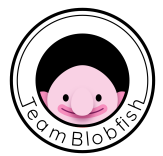
QR6	Usability
<b>Beschreibung</b>	Die vom Team erweiterte Applikation ist für Nutzer verständlich und möglichst leicht zu bedienen.
<b>Begründung</b>	"My-Thai-Star" ist nur dann als Beispiel für devonfw
<b>Erfüllungskriterium</b>	Die Festgelegten Usability Test werden mit den festgelegten Bedingungen entsprechend erfolgreich abgeschlossen



## 5.7 QR7 Ähnlichkeit

Tabelle 65: Qualitätsanforderung 7 - Ähnlichkeit

QR7	Ähnlichkeit
<b>Beschreibung</b>	Komponenten die durch unser Team hinzugefügt wurden folgen dem Stil der Applikation, dass diese sich nicht erheblich vom vorhandenen Design abheben
<b>Begründung</b>	Die gesamte Applikation sollte ein übergreifendes Bild darstellen. Ein einheitliches und Stil-Konformes auftreten trägt zu einer erhöhten Benutzerfreundlichkeit bei
<b>Erfüllungskriterium</b>	Bei den Usability Test ergibt sich, dass die neu eingefügten Elemente sich nicht von den vorherigen abheben und somit die Benutzerfreundlichkeit einschränken



## 6. Qualitätsziele

Die folgende Tabelle beschreibt die zentralen Qualitätsziele der „My-Thai-Star“-Applikation.

Tabelle 66: Qualitätsziele

Qualitätsziel	Motivation und Erläuterung
Sicherheit	Das System soll verhindern, dass jeder Nutzer unberechtigten Zugriff auf Nutzerdaten hat. Jede Nutzerrolle hat seine eigenen Rechte. Nur der Administrator hat Zugriff auf die Nutzerkonten und kann diese verändern.
Benutzbarkeit	Das „My-Thai-Star“-System soll für jeden Kunden leicht zu bedienen sein. Das Kellner-Cockpit und die Administrationsoberfläche sollen für den Kellner und Administrator einfach zu bedienen sein.
Verfügbarkeit	Das System soll jederzeit erreichbar sein und ein Mehrbenutzerbetrieb ermöglichen. Auf tretende Fehler dürfen das System nicht zum Absturz bringen.

## 7. Technische Produktumgebung

### 7.1 Entwicklungsumgebung

Tabelle 67: Randbedingung Entwicklungsumgebung

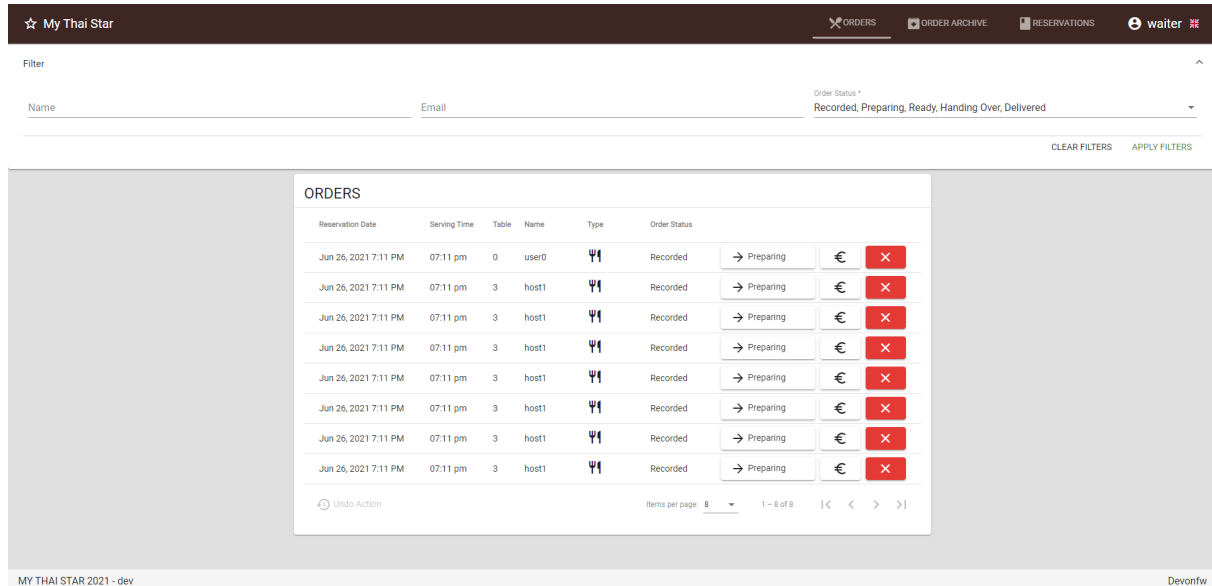
RB1	Für die Projektentwicklung des „My-Thai-Star“-Projekts wird die Entwicklungsumgebung devonfw IDE verwendet
RB2	Die komplette „My-Thai-Star“-Applikation wird in einer Cloud-Deployment Umgebung realisiert



## 7. Produkteinsatz

### 7.1 Weboberfläche

#### 7.1.1 Waiter Cockpit



Filter

Name Email Order Status \* Recorded, Preparing, Ready, Handing Over, Delivered

CLEAR FILTERS APPLY FILTERS

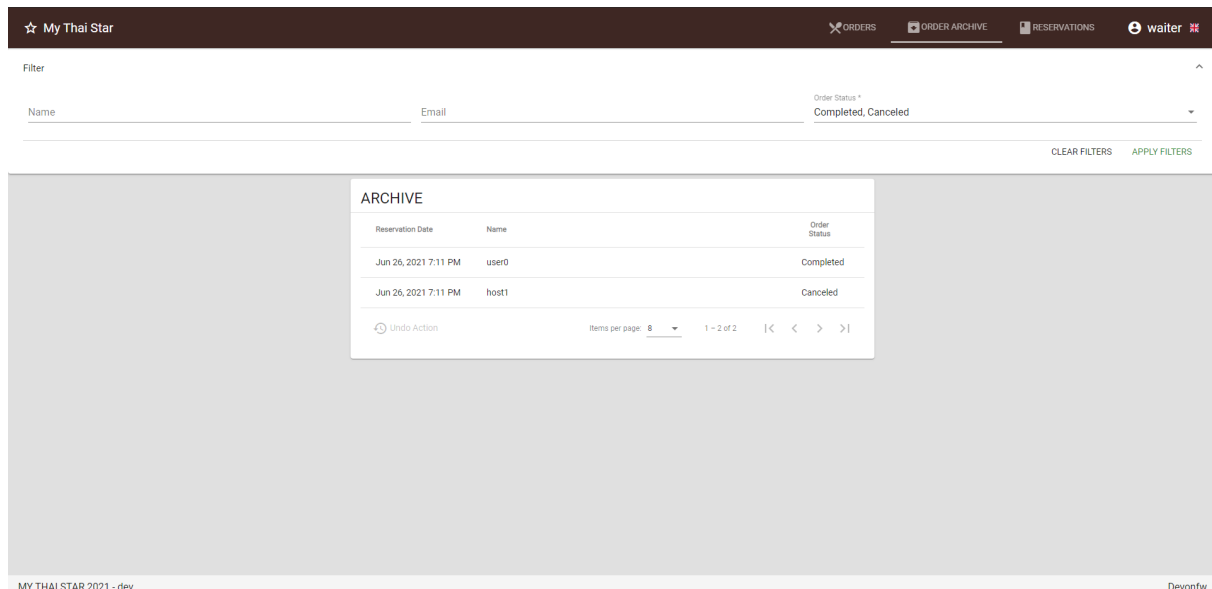
**ORDERS**

Reservation Date	Serving Time	Table	Name	Type	Order Status
Jun 26, 2021 7:11 PM	07:11 pm	0	user0	Ψ	Recorded
Jun 26, 2021 7:11 PM	07:11 pm	3	host1	Ψ	Recorded
Jun 26, 2021 7:11 PM	07:11 pm	3	host1	Ψ	Recorded
Jun 26, 2021 7:11 PM	07:11 pm	3	host1	Ψ	Recorded
Jun 26, 2021 7:11 PM	07:11 pm	3	host1	Ψ	Recorded
Jun 26, 2021 7:11 PM	07:11 pm	3	host1	Ψ	Recorded
Jun 26, 2021 7:11 PM	07:11 pm	3	host1	Ψ	Recorded
Jun 26, 2021 7:11 PM	07:11 pm	3	host1	Ψ	Recorded

Undo Action Items per page: 8 1 - 8 of 8 |< < > >|

MY THAI STAR 2021 - dev Devonfw

Abbildung 3: Waiter Cockpit Bestellungen



Filter

Name Email Order Status \* Completed, Canceled

CLEAR FILTERS APPLY FILTERS

**ARCHIVE**

Reservation Date	Name	Order Status
Jun 26, 2021 7:11 PM	user0	Completed
Jun 26, 2021 7:11 PM	host1	Canceled

Undo Action Items per page: 8 1 - 2 of 2 |< < > >|

MY THAI STAR 2021 - dev Devonfw

Abbildung 4: Waiter Cockpit Bestellsarchiv

☆ My Thai Star

ORDERS

ORDER ARCHIVE

RESERVATIONS

waiter

Filter

Email

Reference number

CLEAR FILTERS

APPLY FILTERS

Reservations

Reservation Date	Email	Reference Number
Jun 26, 2021 7:11 PM	user0@mail.com	CB_20170509_123502555Z
Jun 26, 2021 7:11 PM	user1@mail.com	CB_20170510_123502575Z
Jun 26, 2021 7:11 PM	user2@mail.com	CB_20170510_123502595Z
Jun 26, 2021 7:11 PM	host1@mail.com	CB_20170510_123502655Z
Jun 26, 2021 7:11 PM	host1@mail.com	CB_20170510_123503600Z

Items per page: 8

1 - 5 of 5

<

>

MY THAI STAR 2021 - dev

Devonfw

Abbildung 5: Waiter Cockpit Reservierungen

In Abbildung 3-5 wird das Waiter Cockpit gezeigt. Das Waiter-Cockpit stellt die Benutzeroberfläche für die [Kellner](#) dar. [Use-Case 2](#) "Bestellstatus editieren oder stornieren" ist durch die in Abbildung 3 "Waiter Cockpit Bestellungen" gezeigte Darstellung möglich. [Use-Case 1](#) "Bestellungen archivieren und auf das Bestellarchiv zugreifen" ist durch die in Abbildung 4 "Waiter Cockpit Bestellarchiv" gezeigte Darstellung möglich. Dort können alle Bestellungen eingesehen werden, die einen abgeschlossenen Zustand erreicht haben. Auf die Reservierungen, die durch [Use-Case 14](#) "Essen bestellen über Webanwendung" eingehen, kann der Kellner in Abbildung 5 "Waiter Cockpit Reservierungen" zugreifen.

## 7.1.2 Administrationsoberfläche

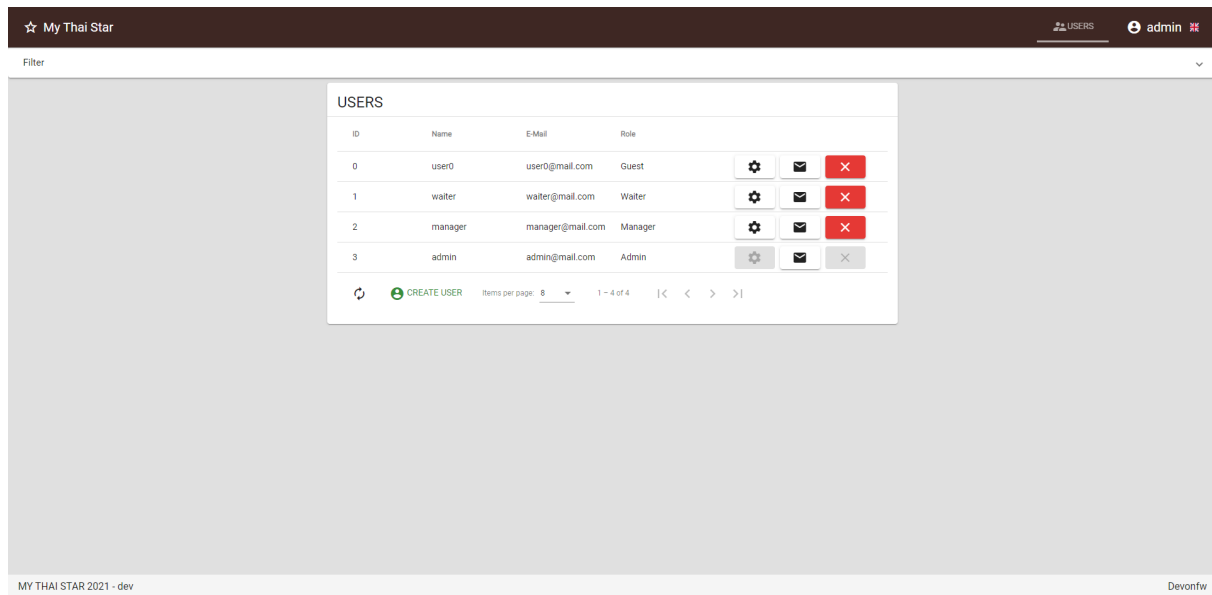


Abbildung 6: Administrationsoberfläche

In Abbildung 6 “Administrationsoberfläche” wird die Ansicht gezeigt, die der [Administrator](#) aufrufen kann. Der Administrator kann hier Benutzerkonten erstellen ([Use-Case 4](#)), löschen ([Use-Case 5](#)), Passwort-Reset-Links versenden ([Use-Case 6](#)), oder Passwörter setzen ([Use-Case 7](#)).

### 7.1.3 Essen bestellen

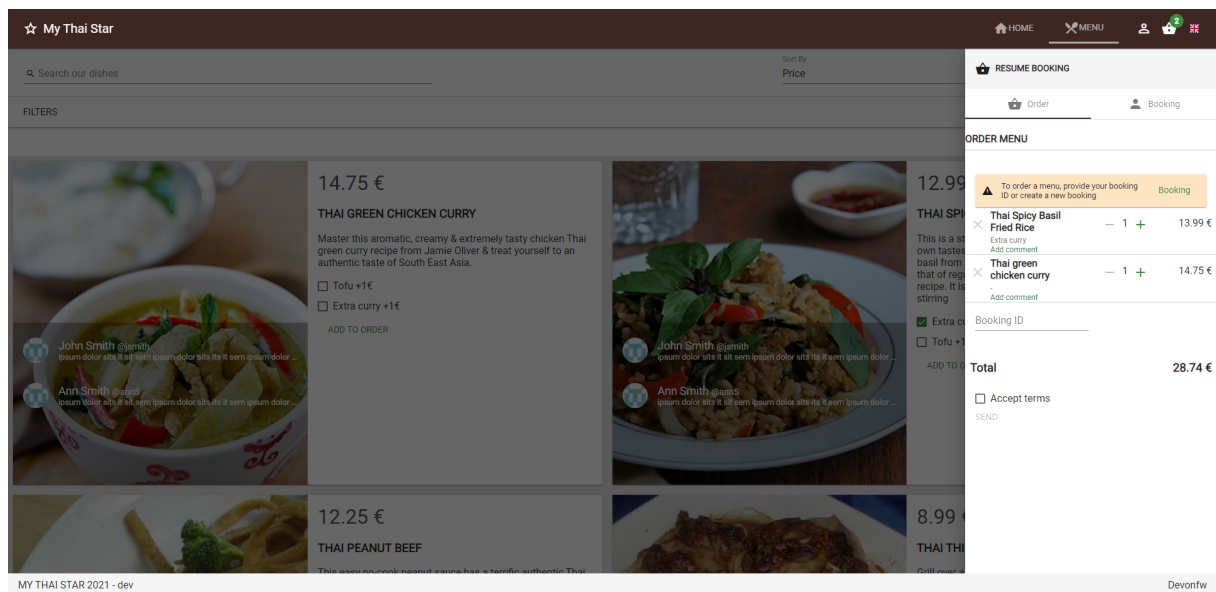


Abbildung 7: Essen bestellen

In Abbildung 7 “Essen bestellen” wird die Ansicht dargestellt, in der der [Kunde](#) sein Essen bestellen kann. Dieser Vorgang wird in [Use-Case 14](#) “Essen bestellen über Webanwendung” beschrieben. Aus dem Menü können Gerichte oder Getränke ausgewählt und der Bestellung hinzugefügt werden.

### 7.1.4 Tisch reservieren

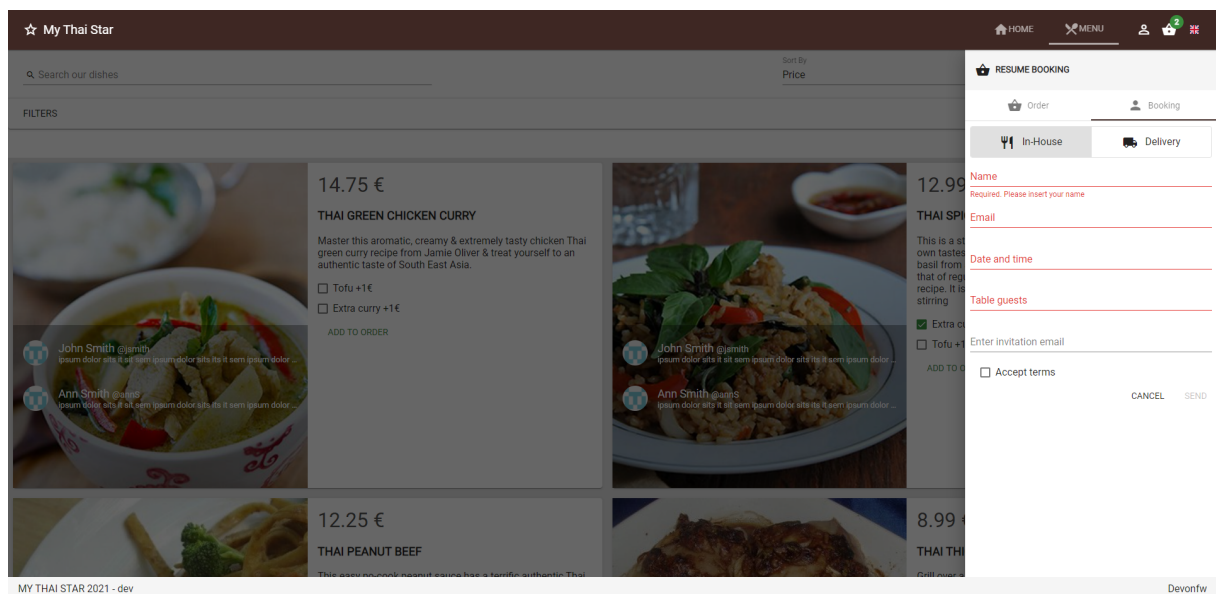


Abbildung 8: Tisch reservieren

Um vor Ort essen zu können, muss der [Kunde](#) einen Tisch reservieren. Dies kann in der Ansicht, die in Abbildung 8 “Tisch reservieren” dargestellt wird, geschehen.

## 7.2 Kontextabgrenzung

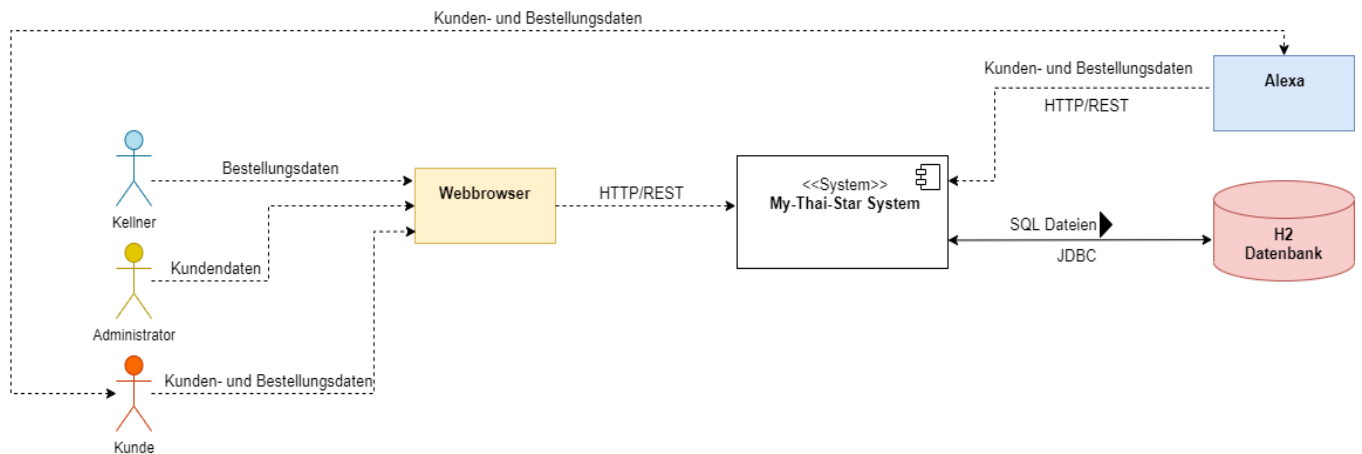


Abbildung 9: Kontextabgrenzung

Diese Abbildung zeigt das fachliche und technische Umfeld des "My-Thai-Star"-Systems.

Genutzt wird die Applikation von Nutzern, die verschiedenen Rollen (Kunde, Administrator oder Kellner) zugeordnet sind. Die Nutzer können alle über den Webbrowser Eingaben tätigen, die im "My-Thai-Star"-System verarbeitet werden und an die H2 Datenbank zur Datenspeicherung weitergeleitet werden.

Eine weitere Schnittstelle, die der Kunde nutzen kann, ist der Sprachassistent "Alexa" von Amazon. Über sprachliche Interaktionen mit Alexa können so die gleichen erforderlichen Daten erfasst werden, die der Kunde sonst über den Webbrowser eingeben müsste. Die Daten werden, wie über den Webbrowser, an das Backend geschickt.

### 7.3 Komponentendiagramm

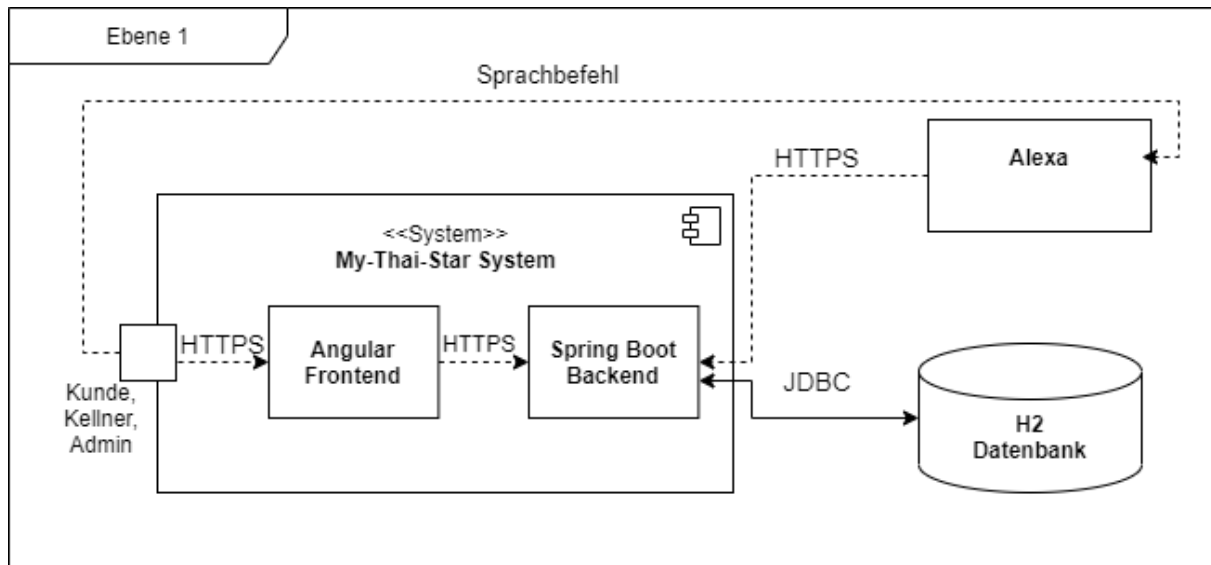


Abbildung 10: Komponentendiagramm Ebene 1

In dieser Abbildung wird das Komponentendiagramm der Ebene 1 dargestellt.

In dieser Ansicht werden die Komponenten des "My-Thai-Star"-Systems und die externen Komponenten Alexa und die H2 Datenbank gezeigt. Auffällig ist die Unterteilung des "My-Thai-Star"-Systems in die Komponenten "Angular Frontend" und "Spring Boot Backend". Das Frontend stellt die Benutzeroberfläche dar, auf der die Inhalte dargestellt werden und mit denen die Nutzer interagieren können. Die Anfragen/Eingaben der Nutzer werden dann vom Frontend an das Backend weitergeleitet, in dem die Daten weiterverarbeitet werden.

## 7.4 Verteilungsdiagramm

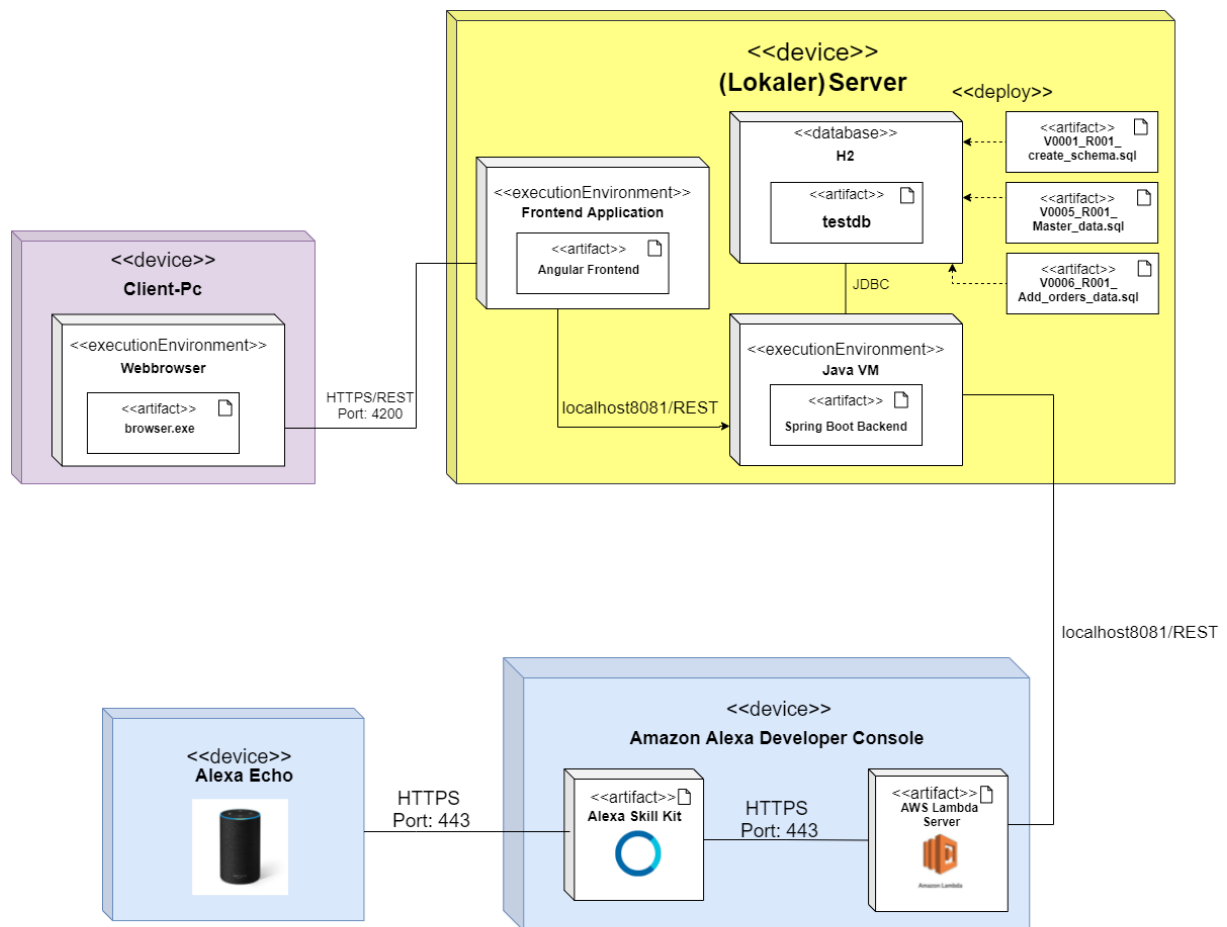
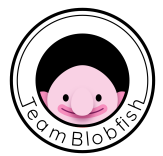


Abbildung 11: Verteilungsdiagramm

Im Verteilungsdiagramm wird dargestellt, welche Software-Elemente von welchen Hardware-Elementen implementiert werden. Der Knoten "Client-PC" stellt die Benutzerschnittstelle dar. In dem Webbrowser kann der Nutzer so seine Eingaben tätigen. Auf dem Server ist das gesamte System integriert. Mit dem Gerät "Alexa Echo" ist es zudem möglich, über den AWS Lambda Dienst mit dem Server zu kommunizieren.



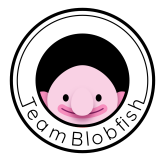
## 8. Auslieferung

Alle unten aufgeführten Dokumente werden dem Kunden über Microsoft Teams geschickt:

- Lastenheft (Erstabgabedatum: 21.04.2021)
- Pflichtenheft (Preview Abgabedatum: 28.04.2021, Erstabgabedatum: 12.05.2021)
- Test Konzept (Erstabgabedatum: 02.06.2021)
- Produkthandbuch (Abgabedatum: 29.06.2021)
- Deployment Dokument (Abgabedatum 1.07.2021)
- Zip-Archiv mit allen Dokumenten (Abgabedatum: 1.07.2021)

Das Projekt wird mit einer Abschlusspräsentation seitens des Teams in der 1. Juliwoche beendet, etwaige Dokumente werden zu diesem Termin auf den aktuellsten Stand gebracht. Des Weiteren wird das Produkt über eine Cloud Lösung, mittels Kubernetes, deployed. Der Quellcode wird dem Kunden am Ende des Projekts (am 1.07.2021) über ein Github-Projekt oder auch Repository zur Verfügung gestellt. Der Kunde kann das Repository klonen und lokal auf dessen Rechner speichern. Anschließend kann der Kunde sich die Quellcodes in einer Entwicklungsumgebung wie Eclipse anschauen und die Anwendung starten.





## 9. Software Glossar

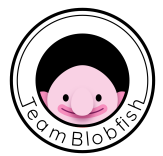
Tabelle 68: Software Glossar

Eclipse	Eclipse ist ein open-source Programmierwerkzeug zur Entwicklung von Software verschiedener Art.
H2 Datenbank	Datenbankserver, der Datenbanksprachen darunter SQL verwendet.

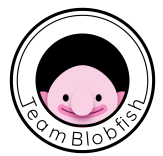
## 10. Glossar

Tabelle 69: Glossar

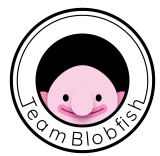
Administrator	Benutzer mit erweiterten Rechten in einem IT-System.
Alexa	Sprachgesteuerte Software von dem Unternehmen Amazon, welches verschiedene Aufgaben erledigen kann oder verschiedene Systeme steuert.
Alexa-Skill	Alexa Skills erweitern Alexa um weitere Funktionen. Dabei handelt es sich um vorgefertigte Szenarien, die auf einen definierten Befehl eine entsprechende Aktion folgen lassen.
Angular	Angular ist ein TypeScript-basiertes Frontend Web Applikations Framework. Es wird von einer Community aus Einzelpersonen und Unternehmen, angeführt durch Google, entwickelt und als Open-Source-Software publiziert.
Artefakt	In UML-Modellen stellen Artefakte Modellelemente dar, die für die physischen Entitäten in einem Softwaresystem stehen. Artefakte stellen physische Implementierungseinheiten dar, z. B. ausführbare Dateien, Bibliotheken,



	Softwarekomponenten, Dokumente und Datenbanken.
Browser	Zentrales Programm in jedem Betriebssystem, das Internet-Inhalte grafisch darstellt und dem Nutzer zugänglich macht.
Bug	Ein Fehlverhalten in einem Computerprogramm.
Client	in Client ist ein Anwendungsprogramm, das auf einem Nutzer-PC und nicht auf einem Server läuft.
Datenbank	Datenbank ist ein System zur elektronischen Datenverwaltung, dazu gehört die Datenspeicherung und Datenbereitstellung.
Devon-Framework/devonfw	Ist ein Softwareentwicklungsplattform, welche eine Reihe von Technologien und Best-Practises enthält und bereitstellt.
Entity-Relationship Diagramm	Das Entity-Relationship Diagramm oder auch Entity-Relationship-Modell dient dazu Objekte aus der realen Welt und ihre Beziehungen darzustellen.
Execution Environment	Der Ausführungsumgebungs-knoten ist Computersystem in einem Geräteknoten. Dabei kann es sich um ein Betriebssystem, eine Java Virtual Machine.
Front-/Backend	Das Backend bezieht sich auf Teile einer Computeranwendung oder eines Programmcodes, die den Betrieb des Programms ermöglichen und auf die ein Benutzer nicht zugreifen kann. Die Schicht oberhalb des Backends ist das Frontend und umfasst die gesamte Software oder Hardware, die Teil einer Benutzerschnittstelle ist.
Geräteknoten	Geräteknoten sind Computing-Ressourcen, die Prozesse verarbeiten und Programme ausführen



	können. Gängige Beispiele für solche Knoten sind PCs, Laptops und Mobilgeräte.
HTTP	Das Hypertext Transfer Protocol (HTTP, englisch für Hypertext-Übertragungsprotokoll) ist ein zustandsloses Protokoll zur Übertragung von Daten auf der Anwendungsschicht über ein Rechnernetz.
JDBC	Java Database Connectivity (JDBC) ist eine treiberbasiert arbeitende Datenbankschnittstelle für Java. Das API erlaubt den Zugriff auf unterschiedliche relationale Datenbanken und ermöglicht die Ausführung von SQL-Anweisungen.
Kellner-Cockpit	Das Kellner-Cockpit ist eine Ansicht, in der Kellner des Restaurants auf alle wichtigen Informationen zu Bestellungen zugreifen können.
Microsoft Teams	Microsoft Teams ist eine von Microsoft entwickelte Plattform, die Chat, Besprechungen, Notizen und Anhänge kombiniert.
REST-API	REST steht für Representational State Transfer, API für Application Programming Interface.  Das Representational State Transfer beschreibt wie verteilte Systeme miteinander kommunizieren können.
Software-Entwicklungsprojekt (SEP)	Das Software – Entwicklungsprojekt ist ein von der Hochschule Mannheim vorgesehenes Semester, in dem Studierende an einem Projekt arbeiten und so ihr Wissen in der Praxis anwenden können.
Spring Boot	Ist ein Open-Source-Framework für Java und hat das Ziel die Komplexität der Java-Plattform zu reduzieren. Es wird häufig genutzt um Webanwendungen in Java zu programmieren.



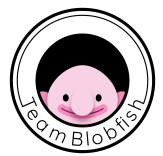
SQL	SQL ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken sowie zum Bearbeiten und Abfragen von darauf basierenden Datenbeständen. Die Sprache basiert auf der relationalen Algebra, ihre Syntax ist relativ einfach aufgebaut und semantisch an die englische Umgangssprache angelehnt.
Use Case	Ein Use Case ist ein Anwendungsfall, der alle möglichen Szenarien, die eintreten können, wenn ein Akteur versucht, mit Hilfe des betrachteten Systems ein bestimmtes fachliches Ziel zu erreichen. Er beschreibt, was inhaltlich beim Versuch der Zielerreichung passieren kann und abstrahiert von konkreten technischen Lösungen. Das Ergebnis des Anwendungsfalls kann ein Erfolg oder Fehlschlag sein.
Wartung	Unter Wartung werden alle Maßnahmen zur vorbeugenden Instandhaltung verstanden.

## 11. Ergänzungen

- Kapitel 2 Grundlegend umgeschrieben
- Kapitel 3.2 und 3.3 wurde überarbeitet
- Überarbeitung der Fit Kriterien für Funktionalen und Nicht-Funktionalen Anforderungen
- Formulierung von Anforderungen wurden geändert.
- Anforderungen wurden hinzugefügt.

Der Bug "Entity references across components #13" wird nicht behoben. Diese Entscheidung traf das Projektteam nach Absprache mit dem Kunden.

Nach Kommunikation kam heraus, dass der Bug je nach Ansicht auf die Architektur relevant ist. Da der Bug viel Zeit benötigt, die gegen Ende des Projekts besser in andere Arbeitspakete gesteckt werden kann, wurde die Entscheidung getroffen, dass der Bug nicht gelöst wird.



## 12. Quellenverzeichnis

Tabelle 70: Quellenverzeichnis

Aufbau des Dokuments und Template/Schema der Tabellen	<a href="https://www.file-upload.net/download-14611165/PflichtenheftV1.2TeamBluebaerFERTIG.pdf.html">https://www.file-upload.net/download-14611165/PflichtenheftV1.2TeamBluebaerFERTIG.pdf.html</a>
Anforderungen des Kunden	<a href="https://www.file-upload.net/download-14611156/2021HSMannheimMyThairStar.pdf.html">https://www.file-upload.net/download-14611156/2021HSMannheimMyThairStar.pdf.html</a>

## 13. Abbildungsverzeichnis

Abbildung 1: Use-Case Diagramm - Kellner und Administrator	11
Abbildung 2: Use-Case Diagramm - Kunde	12
Abbildung 3: Waiter Cockpit Bestellungen	64
Abbildung 4: Waiter Cockpit Bestellungsarchiv	64
Abbildung 5: Waiter Cockpit Reservierungen	65
Abbildung 6: Administrationsoberfläche	66
Abbildung 7: Essen bestellen	67
Abbildung 8: Tisch reservieren	67
Abbildung 9: Kontextabgrenzung	68
Abbildung 10: Komponentendiagramm Ebene 1	69
Abbildung 11: Verteilungsdiagramm	70

## 14. Tabellenverzeichnis

Tabelle 1: Rollen und ihre Rechte/Funktionalitäten	10
Tabelle 2: Use Case Beschreibung – Bestellungen archivieren und auf das Bestellarchiv zugreifen	13
Tabelle 3: Use-Case Beschreibung – Bestellungen verwalten	14
Tabelle 4: Use-Case Beschreibung – Auf Administrationsoberfläche zugreifen	16
Tabelle 5: Use-Case Beschreibung – Benutzerkonten erstellen	17
Tabelle 6: Use-Case Beschreibung – Benutzerkonten löschen	18
Tabelle 7: Use-Case Beschreibung – Passwort-Reset-Link versenden	19
Tabelle 8: Use-Case Beschreibung – Passwörter setzen	20
Tabelle 9: Use-Case Beschreibung – Essen bestellen über Alexa-Inhouse	21
Tabelle 10: Use-Case Beschreibung – Speisekarte abfragen über Alexa-Inhouse	22
Tabelle 11: Use-Case Beschreibung – Servierzeiten festlegen über Alexa-Inhouse	23
Tabelle 12: Use-Case Beschreibung - Essen bestellen über Alexa-Home	24
Tabelle 13: Use-Case Beschreibung – Speisekarte abfragen über Alexa-Home	25
Tabelle 14: Use-Case Beschreibung – Tisch reservieren über Alexa-Home	26
Tabelle 15: Use-Case Beschreibung – Essen bestellen über Webanwendung	27
Tabelle 16: Zusammenfassung der funktionalen Anforderungen mit Prioritäten	28
	77

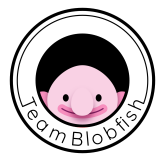
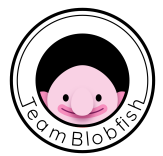


Tabelle 17: Zusammenfassung der nicht-funktionalen Anforderungen mit Prioritäten	30
Tabelle 18: FA – Ansicht von archivierten Bestellungen	31
Tabelle 19: FA – Archivierung von Bestellungen	31
Tabelle 20: FA – Bestellstatus editieren	32
Tabelle 21: FA – Bestellungen stornieren	32
Tabelle 22: FA - Administrationsoberfläche	33
Tabelle 23: FA – Benutzerkonten erstellen	33
Tabelle 24: FA – Benutzerkonten löschen	34
Tabelle 25: FA – Passwort-Reste-Link versenden	34
Tabelle 26: FA – Passwörter setzen	35
Tabelle 27: FA - Bestellungen erhalten Status	35
Tabelle 28: FA – Benutzerrollen zuweisen	36
Tabelle 29: FA – Essen bestellen über Alexa-Inhouse	36
Tabelle 30: FA – Abfragen der Speisekarte über Alexa-Inhouse	37
Tabelle 31: FA – Tisch reservieren über Alexa-Home	37
Tabelle 32: FA – Essen bestellen über Alexa-Home	38
Tabelle 33: FA – Abfrage der Speisekarte über Alexa-Home	38
Tabelle 34: FA – Gast über die Webanwendung bei Reservierung einladen	39
Tabelle 35: FA – Essen bestellen über Webanwendung	39
Tabelle 36: FA – Die Buchungs-ID wird dem Kunden angezeigt	40
Tabelle 37: FA - Servierzeit über Webanwendung bei Bestellung festlegen	40
Tabelle 38: FA – Bugs beheben	41
Tabelle 39: FA – Servierzeiten festlegen über Alexa-Inhouse	41
Tabelle 40: FA – Servierzeiten festlegen über Alexa-Home	41
Tabelle 41: FA – Namen und TableID anzeigen	42
Tabelle 42: FA – Bezahlstatus editieren	42
Tabelle 43: FA – Bestellung filtern	43
Tabelle 44: FA – Benutzerregistration	43
Tabelle 45: FA – Daten der Benutzerkonten editieren	44
Tabelle 46: FA – Die Buchung-ID wird automatisch in die Bestellung eingetragen	44
Tabelle 47: FA – Darstellung auf mobilen Interface	45
Tabelle 48: NFA – Kellner-Cockpit übersichtlich machen	45
Tabelle 49: NFA – Kellner-Cockpit leicht bedienbar machen	46
Tabelle 50: NFA – Administrationsoberfläche übersichtlich machen	46
Tabelle 51: NFA – Administrationsoberfläche leicht bedienbar machen	47
Tabelle 52: NFA – Zuverlässige Alexa-Inhouse Steuerung	47
Tabelle 53: NFA – Alexa-inhouse Steuerung leicht verständlich machen	48
Tabelle 54: NFA – Alexa-Inhouse Steuerung intuitiv	48
Tabelle 55: NFA – Zuverlässige Alexa-Home Steuerung	49
Tabelle 56: NFA – Alexa-Home Steuerung leicht verständlich machen	49
Tabelle 57: NFA – Alexa-Home Steuerung intuitiv	50
Tabelle 58: NFA – Intuitiver Online Reservierungsvorgang	50
Tabelle 59: Qualitätsanforderung 1- Codequalität	51
Tabelle 60: Qualitätsanforderung 2 - Vollständigkeit	51
Tabelle 61: Qualitätsanforderung 3 - Sprache	51
Tabelle 62: Qualitätsanforderung 4 - Modularität	52
Tabelle 63: Qualitätsanforderung 5 - Erweiterbarkeit	52
Tabelle 64: Qualitätsanforderung 6 - Usability	53



Tabelle 65: Qualitätsanforderung 7 - Ähnlichkeit	53
Tabelle 66: Qualitätsziele	54
Tabelle 67: Randbedingung Entwicklungsumgebung	54
Tabelle 68: Software Glossar	56
Tabelle 69: Glossar	56
Tabelle 70: Quellenverzeichnis	59



## 15. Unterschriften

### 15.1 Auftragnehmer

27.04.2021

---

Sven Reinemuth (Datum, Unterschrift)

27.04.2021

---

Exoucia Mukubay (Datum, Unterschrift)

27.04.2021

---

Tony Friedrich (Datum, Unterschrift)

27.04.21

---

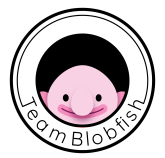
Maximilian Stinner (Datum, Unterschrift)

27.04.2021

---

Florian Braasch (Datum, Unterschrift)





## 15.2 Auftraggeber

---

Dr. Martin Girschick (Datum, Unterschrift)

---

Chris Harsch (Datum, Unterschrift)

---

Cenan Yildirim (Datum, Unterschrift)