Inhalt

[1.1.1 Aufbau des Baukastens 3](#_Toc107593623)

[1.1.1.1 Frame 3](#_Toc107593624)

[1.1.1.2 Layout 4](#_Toc107593625)

[1.1.1.3 Fragment 5](#_Toc107593626)

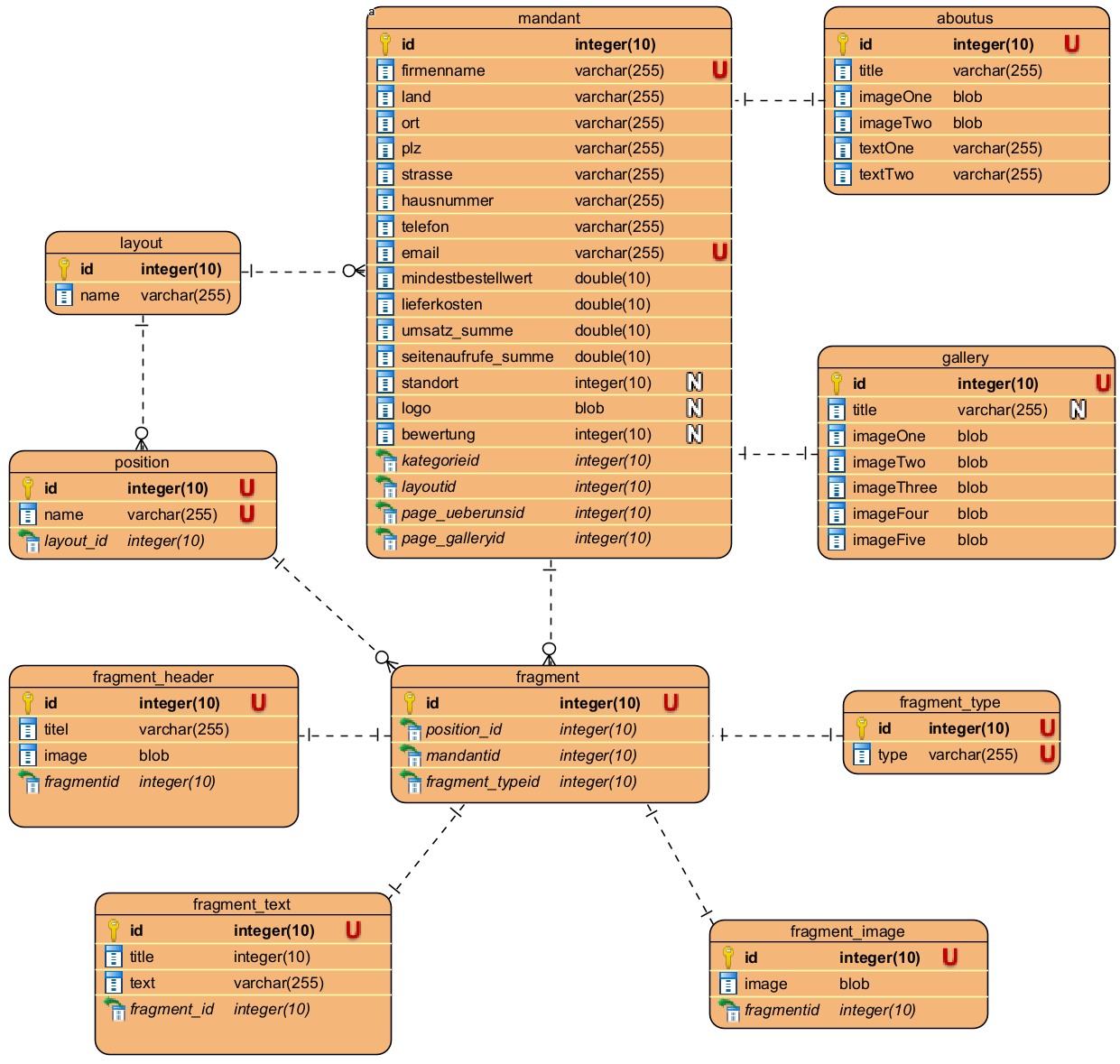
Der Baukasten stellt eine zentrale Komponente des Projektes dar. Über den Baukasten können Restaurant ihre eigenen Seiten erstellen und individuell anpassen. Hierfür müssen viele verschieden Daten gespeichert werden (Abbildung).

Als erstes wird ein Layout benötigt. Das Layout gibt an, welche Vorlage verwendet wird. Diese Vorlage bestimmt die Struktur der Seite (Anzahl der verschiedenen Module, Positionierung dieser Module, Design des Kopfbereiches). Jedes solches Layout besteht aus vielen Positionen. Jede Position enthält ein Modul.

Diese Module gibt es in vielen verschiedenen Formen. Jedes Fragment ist einem Fragment-Typ zugeordnet. Der Fragment-Typ entscheidet, in welcher Form das Modul gespeichert wird und welche Informationen darin enthalten sind.

Im Rahmen dieses Projektes müssen speziell drei Fragment-Typen in der Datenbank gespeichert werden: Fragment-Text, Fragment-Image und Fragment-Header. Fragment-Text ist eines der einfachsten Module. In diesem kann das Restaurant einen beliebigen Titel und einen Text definieren. Das Fragment-Image Modul tauscht den Text mit einem Bild aus. Jedes Layout besitzt ein Fragment-Header Modul, das immer die gleiche Datenstruktur hat, diese aber anders darstellt. Der Header beinhaltet einen Titel, Text und ein Bild.

Des Weiteren kann ein Mandant bis zu zwei zusätzliche Seiten im Baukasten erstellen, eine Bildergalerie- und eine „Über uns“-Seite. In der Bildergalerie sind ein Titel und fünf Bilder enthalten. Die „Über uns“-Seite kann vom Restaurant mit Titel, zwei Texten und zwei Bildern befüllt werden.



IMPLEMENTIERUNG

### Aufbau des Baukastens

Da der Baukasten die Ausgabe von vielen verschiedenen Konfigurationen bewerkstelligen muss, muss dieser sehr modular implementiert werden.

#### Frame

Als Basis des Baukastens dient ein Frame. Dieser Frame ist bis auf ein paar wenige allgemeine Abschnitte (Styles, Navigationsleiste und weiteres) leer und enthält nur ein div, welches durch Thymeleaf mit einem Layout befüllt wird.



Thymeleaf entscheidet mithilfe einiger Informationen des Kontrollers, in welcher der beiden Ansichten: Bearbeitungsmodus und Betrachtungsmodus wir uns befinden. Je nach Modus müssen verschiedene Dinge ausgegeben werden.

In der Ansicht eines Kunden enthält die Navigationsleiste Links, um auf die Start-, „Über uns“, oder Galerieseite zu gelangen. Des Weiteren enthält sie eine Weiterleitung auf die Startseite von Lieferrex und einen Button, um den Warenkorb zu öffnen. In der Ansicht des Restaurants dient die Navigationsleiste einem anderen Zweck. Über diese Leiste kann ein Mandant die „Über uns“, oder Galerieseite erstellen oder entfernen.

Um all diese verschiedenen Varianten zu ermöglichen, wird Thymeleaf verwendet. Anhand von Variablen, die Thymeleaf vom Kontroller erhält, kann entschieden werden, wo welcher Link ausgegeben werden muss. Des Weiteren sind diese Links alle relativ zur aktuellen Position auf der Webseite. Deswegen müssen auch diese von Thymeleaf ausgewertet werden.

In der Abbildung () wird gezeigt wie Thymeleaf entscheidet, welche Links in der Ansicht des Mandanten ausgegeben werden. Die Variablen „AboutUs“ und „Gallery“ enthalten Informationen, ob die zusätzlichen Seiten „Über uns“ und Galerie existieren und ob Links zur Erstellung oder zum Löschen dieser angezeigt wird.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Für den Bearbeitungsmodus werden außerdem weitere Inhalte eingebunden, um alle Funktionen des Baukastens zu ermöglichen (ABBILDUNG). Diese Inhalte beinhalten Modals (Pop-Ups), JavaScript-Code und ein Menü mit allgemeinen Einstellungsmöglichkeiten.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Befinden wir uns nicht im Bearbeitungsmodus, sondern dem Betrachtungsmodus, wird die Ansicht für den Kunden ausgegeben (ABBILDUNG). Diese Ansicht enthält Links zu den Unterseiten des Restaurants, falls diese existieren sollten, und einen Button zum Öffnen des Warenkorbs.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Die mobile Ansicht der Navigationsleiste sieht anders aus, enthält aber die gleiche Logik.

#### Layouts

Zur Auswahl für das Layout gibt es vier Varianten, aus denen das Restaurant wählen kann. Jedes Layout hat eine andere Struktur und Anzahl von Modulen. Ein Layout enthält die verschiedenen Positionen, die mit Modulen befüllt sind. Die erste Position jedes Layouts ist fest definiert als Header. Jedes Layout hat einen eigenen Header, der immer gegeben sein muss. Die Positionen werden in Zeilen und Spalten aufgeteilt. So ist die Position „r1c1“ in der ersten Zeile und Spalte des Layouts.

Bei einer Ausgabe einer solchen Position gibt es drei Möglichkeiten:

1. Fragment existiert und wird ausgegeben
2. Fragment existiert nicht und wir befinden uns im Bearbeitungsmodus
3. Fragment existiert nicht und wir befinden uns im Betrachtungsmodus

Im ersten Fall existiert bereits ein Modul und dieses wird mit Zusatzinformationen (Inhalt, Position und Bearbeitungsmodus) ausgegeben. Im zweiten Fall existiert noch kein Modul an dieser Position und es muss überprüft werden, ob wir uns aktuell im Bearbeitungs- oder Betrachtungsmodus befinden. Im zweiten Fall, dem Bearbeitungsmodus wird ein Platzhalterfragment angezeigt, über welches das Restaurant ein neues Modul anlegen kann. Im letzten Falle, dass wir uns im Betrachtungsmodus aufhalten, wird nichts ausgegeben.

In (ABBILDUNG) wird die Ausgabe eines dynamischen Moduls veranschaulicht. Hier wird je nach Fragmenttyp das richtige Modul geladen.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Bei der Ausgabe des festgelegen Header-Moduls wird auf den entsprechenden Header des Layouts verwiesen und dieser mit den Informationen befüllt (ABBILDUNG).

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

#### Fragmente

Fragmente gibt es in vielen Formen. Jedes Fragment ist für die Aus- oder Eingabe verschiedener Daten zuständig. In TABELLE werden die verschiedenen Typen mit Funktion aufgelistet.

|  |  |
| --- | --- |
| **Fragmenttyp** | **Funktion** |
| Text | Ausgabe eines Titels und einem Text. |
| Image | Ausgabe eines Titels und einem Bild. |
| Header | Kopfbereich der Webseite mit Titel, Text und Bild. |
| Speisekarte | Bestellformular/Menükarte. Listet alle Gerichte des Restaurants. Über dieses Fragment kann ein Kunde bestellen. |
| Google-Maps | Ausgabe eines Titels und einer Google-Map des Restaurant-Standorts. |
| Kontaktinformationen | Stellt die Kontaktinformationen (Telefonnummer, E-Mail und Adresse) des Mandanten dar. |
| Öffnungszeiten | Ausgabe einer Liste der Öffnungszeiten des Restaurants. |
| Add | Fragment zur Bedienung des Baukastens. Wird verwendet, um ein neues Modul einzufügen. |

Jedes Fragment hat den gleichen Aufbau, nur der Inhalt ändert sich je nach Typ. In ABBILDUNG wird als Beispiel das einfache Text-Fragment gezeigt. Alle Fragmente, mit Ausnahme des Header-Fragments hat außerdem einen Knopf, der nur im Bearbeitungsmodus angezeigt wird (wieder über die „edit“ Variable ausgewertet), um ein bestehendes Modul zu löschen und mit einem anderen auszutauschen. Das Header-Fragment kann nicht gelöscht, aber angepasst werden.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Das Image-Fragment sieht ähnlich aus, der Text wurde aber mit einem Bild ausgetauscht. Hierbei muss darauf geachtet werden, dass vom Kontroller kein Pfad zum Bild, sondern ein Base64-String übergeben wird. Um diesen String als Bild darzustellen, muss in der „src“ des Bildes darauf geachtet werden (ABBILDUNG).

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Das Fragment für die Speisekarte fügt nicht nur eine einfache Oberfläche wie die anderen Fragmente hinzu, sondern aus weiter Funktionalität, die für den Bestellvorgang benötigt wird. Das Speisekarte-Fragment liefern JavaScript-Code zur Verwaltung des Warenkorbs und einer Oberfläche für den Warenkorb mit.

In ABBILDUNG wird der Ablauf dargestellt, wie ein Gericht dem Warenkorb hinzugefügt wird. Zuerst wird der Warenkorb Cookie ausgelesen, ist dieser leer, wird er neu angelegt. Danach wird aus den verschiedenen Datenfeldern des Gerichtes der Input des Kunden ausgelesen. Es wird geprüft, ob ein Gericht mit gleicher ID und Namen bereits im Warenkorb ist. Wenn ja wird die Anzahl dieses Gerichtes erhöht. Wenn nein wird es neu erstellt. Das neue Cookie wird gespeichert und die Darstellung auf der Webseite neu geladen.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Die „updateWarenkorb“ Funktion (ABBILDUNG) liest das Cookie aus und erstellt eine Tabelle daraus. Diese Tabelle wird im Warenkorb angezeigt.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Um ein Gericht wieder aus dem Warenkorb zu löschen, wird die Funktion „removeGericht“ ausgeführt (ABBILDUNG). Das Gericht mit gleicher ID und Anmerkung wird aus der Liste gelöscht, das Cookie gespeichert und der Warenkorb aktualisiert.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

#### Modals