

Inhaltsverzeichnis

1 Projekt - SS2023 - Gastrobit.de	1
2 Ausarbeitung: Entwicklung einer Alternative zu Lieferando	1
2.1 Ermittlung der Anforderungsaufnahme	1
2.2 2. Identifikation der Anwendungsfälle des Softwareprodukts	1
2.3 Entwurf einer Benutzungsoberfläche für das Produkt	3
2.4 Herleitung eines Datenmodells	3
2.4.1 Restaurants	3
2.4.2 Custom Domains	4
2.4.3 Orders	4
2.4.4 Datenbankdiagramm	5
2.4.5 Row-Level Security	5
2.5 Beschreibung der zentralen eingesetzten Algorithmen	6
2.5.1 Liefergebiet	6
2.5.2 Schemas	6
2.5.3 Preise ausrechnen	7
2.6 Geplante Abnahmetests zur Validierung der Anforderungen	8
2.7 Geplante Inbetriebnahme auf technischer Ebene	8
2.7.1 Optionale Anleitung für Self-Hosting	8
2.8 Einführung der Nutzung des Produkts durch den Endnutzer	10
2.8.1 Registrierung und Login	10
2.8.2 Restaurantübersicht	11
2.8.3 Restaurantverwaltung	12
2.8.4 Speisekarte	17
2.8.5 Bestellen	24
2.8.6 Bestellungen abarbeiten	29
2.8.7 Lieferstatus	30
2.9 Quellen	31

1 Projekt - SS2023 - Gastrobit.de

Name	Matrikelnummer	E-Mail
Marc-Alexander Richts	10059445	richts.marc-alexander@fh-swf.de

2 Ausarbeitung: Entwicklung einer Alternative zu Lieferando

2.1 Ermittlung der Anforderungsaufnahme

Die Anforderungen für unsere Alternative zu Lieferando wurden durch eine Kombination aus Marktforschung und einem Gastronomeninterview ermittelt. Die Hauptanforderungen sind:

- Benutzerfreundliche Oberfläche für den ENdkunden
- Responsive Design für mobile Endgeräte beim Kaufvorgang durch Endkunden
- Einfache Bestellprozesse für den Endkunden
- Bezahlungsfunktion für den Endkunden
- Restaurantverwaltung durch den Besitzer (Speisekarte, Erreichbarkeit über eigene Domain/*.gastrobit.de-Subdomain, Auswahl eines Designs, Festlegung des Liefergebietes, etc.)
- Tracking und Verwaltung des Bestellvorgangs
- Niedrige Kommissionen für Gastronomen

2.2 2. Identifikation der Anwendungsfälle des Softwareprodukts

Die Anwendungsfälle für unser Produkt wurden identifiziert und in einem Use Case Diagramm dargestellt.

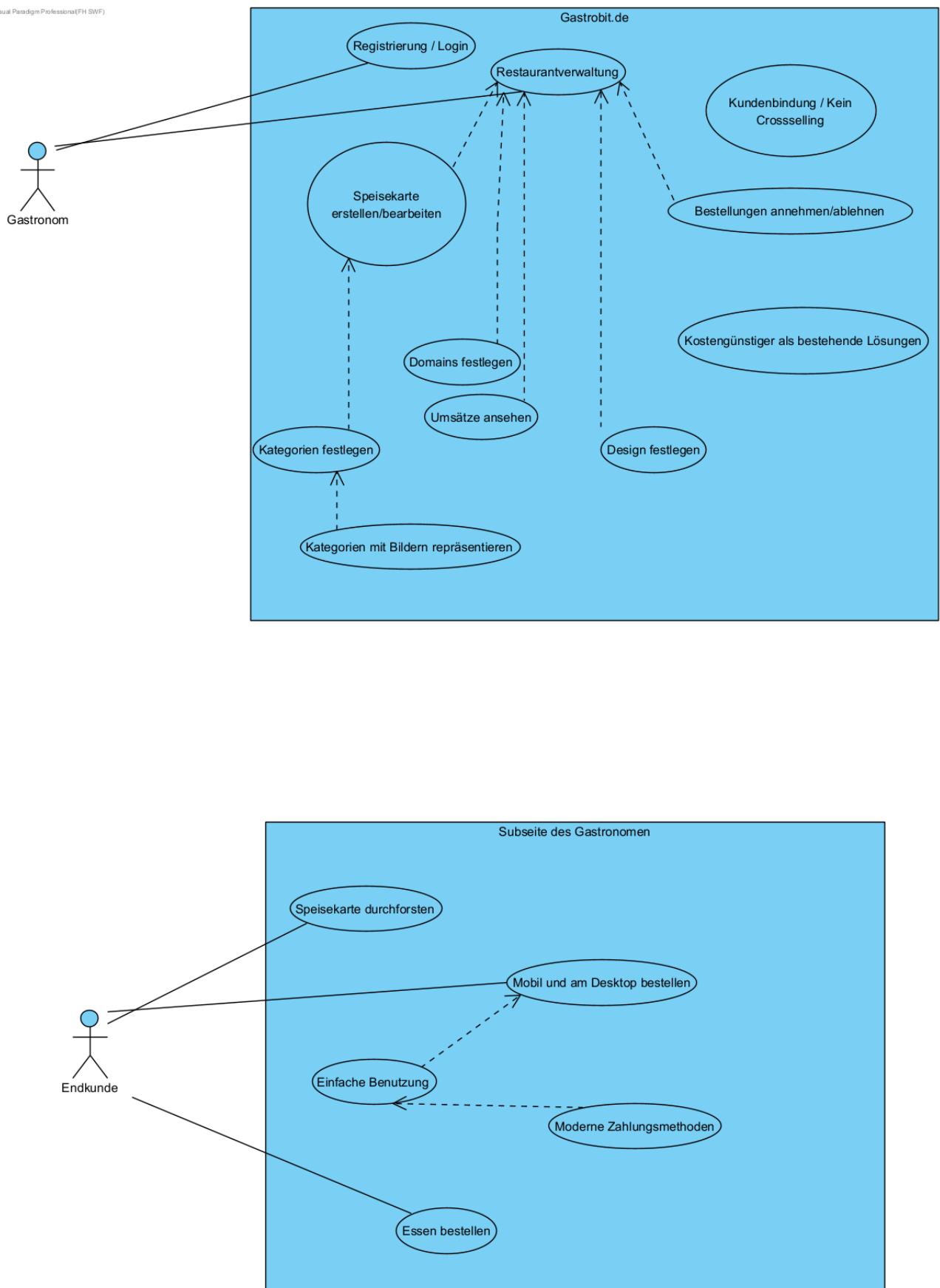


Figure 1: Use Case Diagramm für Gastronomen und Endnutzer

Die Hauptanwendungsfälle sind:

- Registrierung, Login, Passwort zurücksetzen sowie E-Mail ändern für Gastronomen

- Hinzufügen und Verwalten von Restaurants durch den Gastronomen
- Übersicht über Umsätze von den einzelnen Restaurants
- Ändern von Gerichten auf der Speisekarte durch den Gastronomen (Extras (z.B. Käse, Zwiebeln, etc.), Preise, Varianten (große Pizza, kleiner Pizza), Gerichte kategorisieren, etc.)
- Bestellung von Gerichten durch Endkunden
- Bezahlung der Bestellung durch Endkunden
- Verwaltung der Bestellung durch den Gastronomen

2.3 Entwurf einer Benutzungsoberfläche für das Produkt

Die Benutzungsoberfläche wurde mit Wireframes entworfen, um eine benutzerfreundliche und intuitive Navigation zu gewährleisten. Die Hauptelemente der Benutzungsoberfläche sind:

- Startseite mit Übersicht über Restaurants die im Besitz des Gastronomen sind
- Verwaltungsfunktion für Restaurants (festlegen von Domains, Liefergebiet, etc.)
- Restaurantseite mit Speisekarte und Bestellfunktion und Bezahlmöglichkeit für Endkunden

Die Wireframes sind im Anhang A zu finden.

2.4 Herleitung eines Datenmodells

Das Datenmodell wurde auf Basis der identifizierten Anforderungen und Anwendungsfälle erstellt. Es besteht aus folgenden Tabellen:

- Restaurants
- Custom Domains
- Orders
- Users (Verwaltet durch Supabase)

2.4.1 Restaurants

Diese Tabelle speichert alle Restaurants, die von Gastronomen erstellt wurden. Sie enthält alle wichtigen Informationen, wie beispielsweise den Namen des Restaurants, die User-ID des Gastronomen, die ID des Stripe-Subaccounts (Connect-Express-Account) sowie die Speisekarte als JSON und das Liefergebiet als Koordinatenarray (Längengrad und Breitengrad), welches ein geschlossenes Polygon ergibt.

2.4.1.1 Warum wird die Speisekarte als JSON gespeichert?

Hier wäre die Chance gewesen, um eine weitere Tabelle zu erstellen und eine 1:n-Beziehung zwischen Restaurant und Gerichten zu erstellen. Allerdings ist die Speisekarte ein sehr komplexes Objekt, welches viele Eigenschaften hat und mit den Extra-Attributen, Varianten und verschiedenen Preisen wären die JOINS sehr komplex geworden. Zudem wird die Speisekarte *immer* komplett an den Nutzer ausgegeben, und es werden keine Teilabfragen benötigt. Daher war es für uns sinnvoller, die Speisekarte als JSON zu speichern.

2.4.1.2 Polygon? Liefergebiet?

Wir hatten mehrere Ansätze um ein Liefergebiet zu definieren.

1. **Radius um das Restaurant.** Dies wäre der simpelste Ansatz gewesen, jedoch fallen hier schnell die Limitierungen dieser unflexiblen Lösung auf: Nehmen wir einfach die Stadt Hamburg, die durch die Elbe horizontal getrennt ist. Ein Gastronom möchte beispielsweise in Hamburg nur den nördlichen Teil der Elbe bedienen, da eine Fahrt durch den Elbtunnel für den Lieferanten zu lange dauert. Ein primitiver Radius würde dies nicht ermöglichen.
2. **Postleitzahlen.** Ein ähnliches Gebiet wie bei dem Radius: Postleitzahlengebiete sind teilweise sehr groß und können Gebiete abdecken, die durch große Hindernisse (wie der oben beschriebene Elbtunnel) geographisch getrennt sind. Außerdem war es mir nicht möglich, eine Datenbank mit Postleitzahlen zu finden, die auch die Koordinaten der Postleitzahlen enthält.
3. **Polygon.** Ein Polygon ist ein geschlossenes Gebiet, welches durch Koordinaten definiert wird. Dies ermöglicht es, sehr komplexe Gebiete zu definieren, die auch natürliche Hindernisse wie Flüsse oder Autobahnen berücksichtigen. Beim Bestellprozess wird nun OpenStreetMaps.com mit der Adresse des Kunden kontaktiert, um die Lieferadresse zu Koordinaten aufzulösen, und dann wird mit der Hilfe des “Point Inclusion in Polygon Test”-Algorithmus festgestellt, ob sich dieser Punkt im Polygon befindet. Siehe dazu den Algorithmenabschnitt in Kapitel 5. **Dies ist die finale Lösung, die in diesem Projekt verwendet**

wurde, da sie die größte Flexibilität bietet und besser an die spezifischen Bedürfnisse der Gastronomen angepasst werden kann.

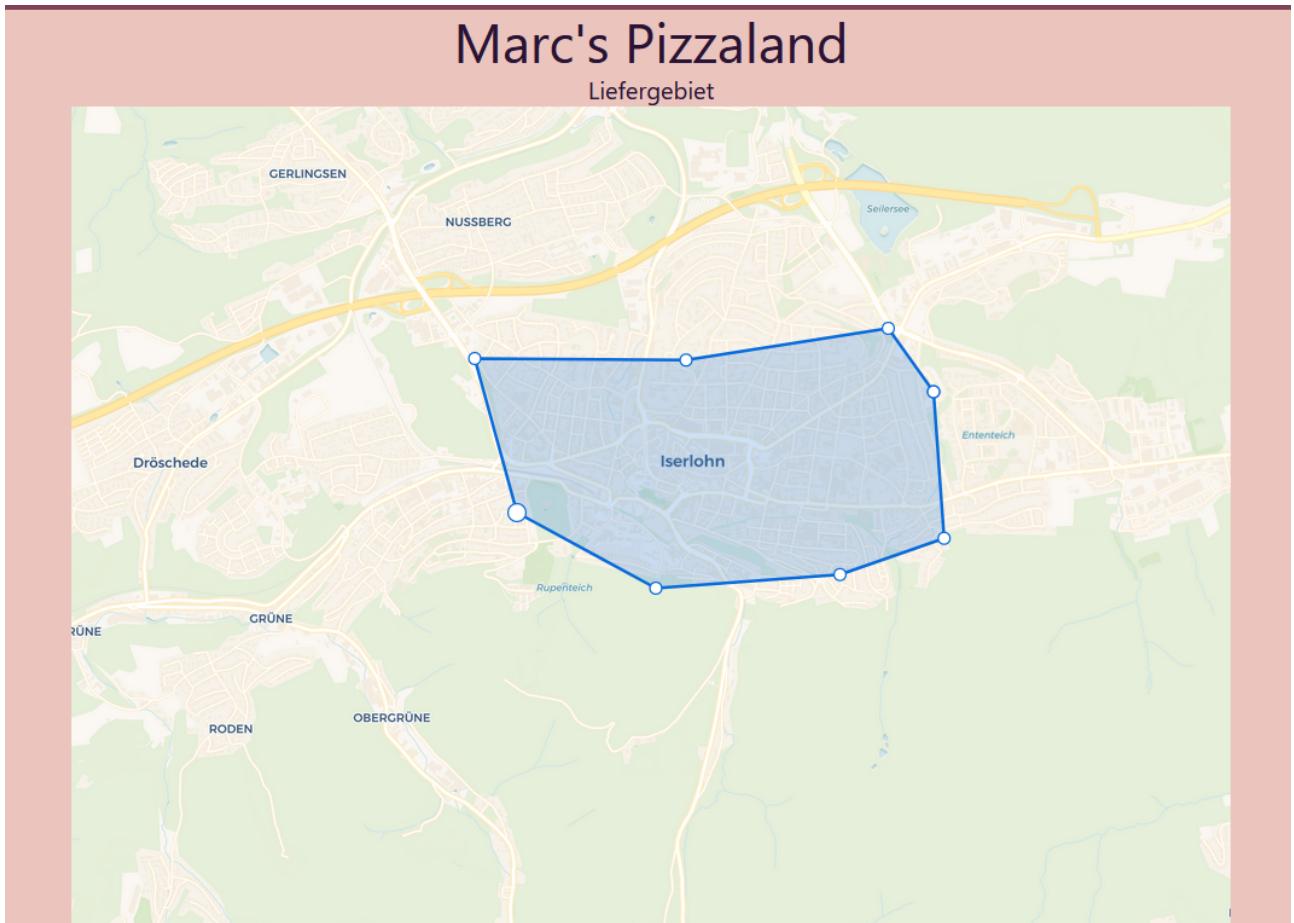


Figure 2: Beispielliefergebiet von “Marc’s Pizzaland” welches den Kern von Iserlohn abdeckt, jedoch nicht die äußeren Stadtteile.

2.4.2 Custom Domains

Eine primitive Tabelle, welche alle benutzerdefinierten Domains der Gastronomen speichert. Beim hinzufügen oder entfernen einer Domain wird ein Nameservereintrag bei unserem Hostinganbieter AWS (via Vercel.com) ebenso hinzugefügt oder entfernt. Diese Tabelle wird für jeden Besuch der Website aufgerufen, um zu prüfen, ob die Domain des Besuchers mit einer Domain eines Restaurants übereinstimmt. Wenn dies der Fall ist, wird der Besucher auf die Restaurantseite weitergeleitet. Dieser Mechanismus ist ähnlich wie *VHosts* bei Apache oder NGINX.

2.4.3 Orders

Diese Tabelle speichert alle Bestellungen, die über die Plattform getätigten wurden. Jede Bestellung ist mit einem Restaurant verknüpft und enthält Informationen wie die Bestelldetails, den Zahlungsstatus, den Bestellstatus (“Ausgeliefert”, “In Bearbeitung”, “Offen”, “Abgelehnt”; *im Falle einer Ablehnung bekommt der Kunde sein Geld zurück*), den Checkout-Link und die Lieferadresse des Kunden.

2.4.4 Datenbankdiagramm

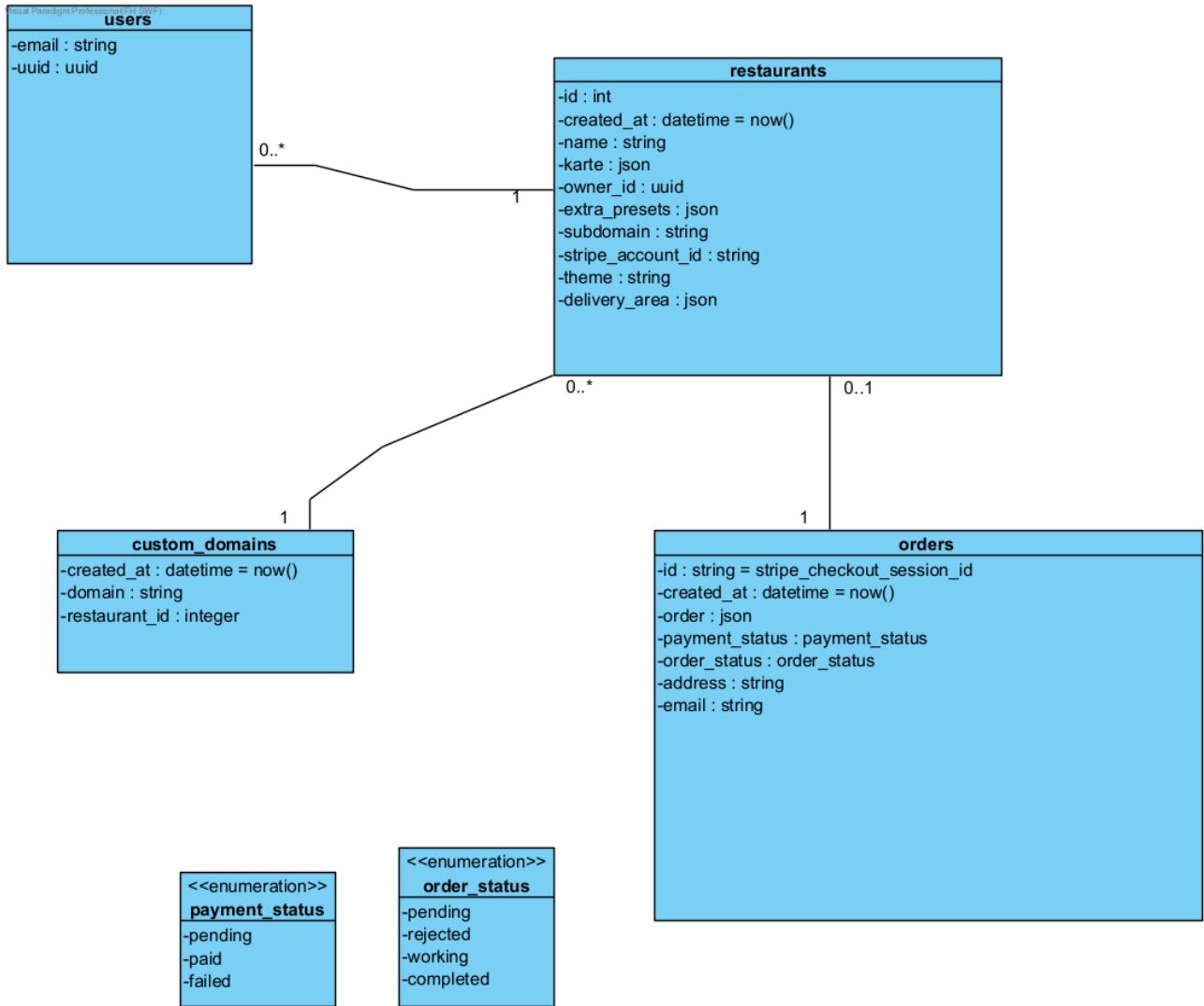


Figure 3: Datenbankdiagramm

2.4.5 Row-Level Security

Da wir uns für eine Serverless-Architektur aus Kosten- und Skalierungsgründen, sowie aus Gründen der Einfachheit entschieden haben, haben wir uns für Supabase als Datenbankanbieter entschieden. Supabase bietet eine primitive PostgreSQL-Datenbank mit Row-Level Security an, welche es ermöglicht, die Datenbank auf Zeilenebene zu schützen. Dies bedeutet, dass wir die Datenbank beispielsweise so konfigurieren können, dass ein Gastronom nur auf seine eigenen Restaurants updaten kann, und nicht auf die Restaurants anderer Gastronomen sehen kann. Die Standardpolicy ist hier “Least Privilege”, das bedeutet dass niemand operationen ohne explizite Erlaubnis per policy erstellen kann.

Hier sind zwei Beispieldolicies for unsere Orders-Tabelle:

```
ALTER TABLE orders ENABLE ROW LEVEL SECURITY;
```

```
create policy "Nur Restaurantbesitzer können Bestellungen updaten"
  on orders
  for update using (
    auth.uid() IN (
      SELECT restaurants.owner_id
      FROM restaurants
      WHERE (restaurants.id = orders.restaurant_id)
      -- Mir ist der Tippfehler bewusst; jedoch habe ich zu weit programmiert um ihn zu diesem Zeitpunkt
    )
```

```

);
create policy "Jeder kann Bestellungen hinzufügen"
on orders
for insert WITH CHECK (
    true
);

```

Die vollständigen RLS-Policy-Expressions sind im Ordner doku/policies/policies.sql zu finden zu finden.

2.5 Beschreibung der zentralen eingesetzten Algorithmen

2.5.1 Liefergebiet

Um festzustellen ob ein Punkt in einem Polygon liegt, wird der “Point Inclusion in Polygon Test”-Algorithmus verwendet. Dieser ist von W. Randolph Franklin veröffentlicht worden, und wurde integriert um zu prüfen, ob sich die Lieferadresse des Kunden im Liefergebiet des Restaurants befindet.

Siehe dazu https://wrfranklin.org/Research/Short_Notes/pnpoly.html

Diesen hatte ich zu Anfang selbst implementiert, jedoch ist mir bei einem späteren Wechsel von Leaflet.js zu @freenow/react-polygon-editor (das Tool, welches zum interaktiven zeichnen von Polygonen verwendet wird), aufgefallen, dass genau dieser gleiche Algorithmus ebenfalls in diesem Paket verwendet wird. Daher habe ich mich entschieden, meine eigene Implementation zu ersetzen, da der von FreeNow einfacher zu lesen ist.

2.5.2 Schemas

Da wir in unserer Datenbank ein JSON-Objekt speichern, müssen wir sehr viel auf Server- und Clientseite validieren.

Dies bedeutet, dass wir JSON-Schemas mit der Library “Zod” erstellen, um die Daten auf Korrektheit zu validieren. Hierfür müssen wir sehr generisch bleiben, da wir nicht wissen, welche Gerichte, Extras, Varianten, etc. der Gastronom anbietet. Daher habe ich Schemas aus Subschemas erstellt, welche erlauben, verschiedene Kompositionen zu erstellen:

```

// Das äußerste Blatt eines JSON-Schemas sind die Extras. Ein Extra hat einen Namen und einen Preis.
export const extra = z.object({
    name: z.string(),
    preis: z.number(),
});

// Ein Extras-Array besteht aus einem Objekt, welches den Namen der Extrakategorie, den Typ (oneOf oder
export const extras = z.array(
    z.object({
        name: z.string({
            description: "Der Name des der Extrakategorie, z.B. 'Ihr Salatdressing'",  

        }),
        typ: z.enum(["oneOf", "manyOf"]), {
            description:  

                "Der Typ der Extrakategorie, z.B. 'oneOf', wenn nur eine Option ausgewählt werden kann",  

        }),
        items: z.array(extra, {
            description: "Die verfügbaren Extras für ein Gericht",
        }),
    })
);

// Ein Preis existiert für Variationen eines Gerichts. Ein Preis hat einen Namen (z.B. "klein", "groß",
const preis = z.object({
    name: z.string({
        description:  

            "Der Name des Preises, z.B. 'klein (18cm)', 'mittel', 'groß (30cm)'",  

    }),
    preis: z.number({
        description:
    })
});

```

```

        "Der Preis in Euro des Gerichts für die angegebene Größe, z.B. 5.5",
    },
});

// Ein Gericht besteht aus einer ID, einer Überschrift, einer Unterschrift, mindestens einer Preisvariante
export const gericht = z.object({
    id: z.string().or(z.number()),
    ueberschrift: z.string(),
    unterschrift: z.string(),
    preise: z
        .array(preis, {
            description: "Die verfügbaren Größen und Preise für das Gericht",
        })
        .min(1),
    extras: extras.optional(),
});
}

// Eine Kategorie besteht aus einer Überschrift, einer ID, einem Headerbild und einer Liste von Gerichten
export const category = z.object({
    headerUrl: z.string().optional(),
    id: z.string().or(z.number()),
    name: z.string(),
    gerichte: z.array(gericht),
});
}

// Eine Speisekarte besteht aus einer Liste von Kategorien.
export const categories = z.array(category);

```

Diese Struktur erlaubt uns, sehr einfach den Zustand unserer Speisekarte zu verwalten, und die Daten auf Korrektheit zu validieren. Außerdem macht es die Arbeit mit dem Zustandsautomaten des Warenkorbes wesentlich leichter, da diese Struktur uns erlaubt, Autocomplete von VSCode zu verwenden.

2.5.3 Preise ausrechnen

Hierbei handelt es sich um nichts wirklich complexes, jedoch ist es wichtig, dass die Preise korrekt berechnet werden. Hierbei handelt es sich um einen einfachen Algorithmus, der die Preise der Variante, und der Extras addiert.

Hier ist es wichtig zu beachten, dass es wie oben beschrieben “ManyOf” und “OneOf”-Extras gibt.

Eine Pizza bietet beispielsweise die Auswahl *einer* Pizzasoße (z.B. “Tomate” ODER “Hollondaise”), jedoch die Auswahl von *mehreren* Belägen (z.B. “Käse”, “Zwiebeln”, “Paprika”, etc.).

```

const calcPrice = () => {
    let price =
        gericht.preise.find((preis) => preis.name === variante)?.preis ?? 0;

    // add the price of the extras
    Object.keys(attribute).forEach((key) => {
        const extra = gericht.extras?.find((extra) => extra.name === key);
        if (extra) {
            if (extra.typ === "oneOf") {
                const item = extra.items.find((item) => item.name === attribute[key]);
                if (item) {
                    price += item.preis;
                }
            } else {
                // manyOf
                const items = attribute[key] as number[];
                items.forEach((index) => {
                    const item = extra.items[index];
                    if (item) {
                        price += item.preis;
                    }
                })
            }
        }
    });
}

```

```

        });
    }
};

return price;
};

```

2.6 Geplante Abnahmetests zur Validierung der Anforderungen

Die Abnahmetests werden durchgeführt, um die Funktionalität des Produkts zu überprüfen und sicherzustellen, dass alle Anforderungen erfüllt sind. Sie umfassen unter anderem:

- Testen der Registrierungs- und Login-Funktionen
- Testen der Restaurantverwaltung durch den Gastronomen
 - Hinzufügen von Restaurants
 - Erstellen des Stripe-Subkontos
 - Festlegen des Liefergebietes
 - Hinzufügen einer benutzerdefinierten Domain
 - Hinzufügen einer *.gastrobit.de-Subdomain
 - Auswahl eines Designs
 - Hinzufügen von Gerichten
 - Testen des Stripe-Dashboards des Subkontos
- Testen des Bestellprozesses und der Bezahlung für den Endkunden
- Testen des Bestellstatus und der Verwaltung der Bestellung durch den Gastronomen

2.7 Geplante Inbetriebnahme auf technischer Ebene

Die Inbetriebnahme umfasst die Installation und Konfiguration des Servers, die Einrichtung der Datenbank und die Bereitstellung der Anwendung im Internet. Wir liefern bereits eine `.env`-Datei mit, welche die Konfiguration für die Datenbank, Stripe, und Vercel enthält. Diese muss lediglich mit den eigenen Daten gefüllt werden, kann jedoch zu Entwicklungszwecken auch weiterhin mit unseren Daten verwendet werden.

2.7.1 Optionale Anleitung für Self-Hosting

Es werden folgende Accounts benötigt:

- Vercel.com
- Supabase.com
- Stripe.com (verifiziertes Konto)

Außerdem werden Grundkenntnisse in Git benötigt.

2.7.1.1 Vercel.com / AWS

Vercel.com / AWS bietet eine günstige Plattform, solange man unter den Limits bleibt. Bei unter 200M Requests pro Monat können wir unseren Dienst auf der Serverlessplattform von Vercel kostenlos betreiben. Dies ist ein großer Pluspunkt für uns, da wir so die Kosten für den Betrieb des Dienstes sehr gering halten können. Falls Gastrobit.de kommerziell wird, können wir auch ohne große Probleme horizontal skalieren.

1. Erstellen Sie ein neues Projekt auf Vercel.com
2. Kopieren Sie alle Projektdateien in eine Git-Repository, und klicken Sie auf [vercel.com](#) auf “Import Project”, und wählen Sie z.B. von Github.com, Gitlab.com o.ä. ihre Git-Repository aus. Sie können den Namen der Branch durch “GIT_BRANCH_FOR_DOMAINS” definieren, falls Sie eine andere Branch als “main” verwenden möchten.
3. Kopieren Sie “PROJECT_ID_VERCEL”, “TEAM_ID_VERCEL” aus den nun angezeigten Projekteinstellungen in die `.env`-Datei im Root-Verzeichnis dieses Projektes.
4. Erstellen Sie auf Vercel.com ein neues Token, und kopieren Sie es als “AUTH_BEARER_TOKEN” in die `.env`-Datei.
5. Nun wird Vercel automatisch nach jedem Git-Commit die Anwendung neu bauen und deployen. Durch das Bearer-Token ist es unserer Anwendung außerdem möglich, die Domains zu verwalten und DNS-Einträge für die Gastronomen zu verwalten.

Wenn Sie statt [vercel.com](#) direkt auf AWS hosten wollen, ist das ebenso über die Vercel-AWS-Extension möglich.

2.7.1.2 Supabase.com

Supabase bietet eine günstige Plattform, solange man unter den Limits bleibt. Bei unter 1GB DB-Größe können wir unseren Dienst auf der Postgresql-Datenbank von Supabase kostenlos betreiben. Dies ist ein großer Pluspunkt für uns, da wir so die Kosten für den Betrieb des Dienstes sehr gering halten können. Falls Gastrobit.de kommerziell wird, können wir auch ohne große Probleme mit Read-Replicas skalieren; da die meisten Operationen auf unsere Datenbank reads sind.

1. Erstellen Sie ein neues Projekt auf Supabase.com
2. Wählen Sie eine Region aus, die zu ihren Endnutzern physisch nah ist und aus Datenschutzsicht sinnvoll ist, z.B. Frankfurt in Deutschland.
3. Navigieren Sie zu den Projekteinstellungen und kopieren Sie folgende Daten in Ihre .env-Datei:
 1. Die Projekt-URL als NEXT_PUBLIC_SUPABASE_URL
 2. Anon-Key als NEXT_PUBLIC_SUPABASE_ANON_KEY
 3. Service-Role/Secret-Key als SUPABASE_SERVICE_KEY
4. Falls Sie unser Produkt selbst weiterentwickeln möchten, können Sie in Supabase ein eigenes Access-Token definieren. Dies erlaubt es Typen aus der Datenbank zu exportieren. Wir liefern dafür das script `npm run types` mit, welches automatisch die neusten Typedefinitions aus der Datenbank kopiert. Dies macht die Entwicklung sehr viel leichter. Das Access-Token muss als SUPABASE_ACCESS_TOKEN in der .env-Datei definiert werden.
4. Für das Erstellen des Datenbankschemas gibt es zwei Möglichkeiten:
 1. Verwenden Sie das Tool `pg_restore -h db.<project-id>.supabase.co -U postgres -W -d <database-name> -1 -v "POSTGRES_BACKUP"`, um die Tabellen zu erstellen. (Empfohlen, und wesentlich weniger Arbeit). Die Datei POSTGRES_BACKUP finden Sie im Ordner über dem Projekt.
 2. Erstellen Sie eine neue Datenbank, und führen das Tabellenscript von Hand aus
 1. Installieren Sie die postgres-SQL-Shell, z.B. mit `sudo apt install postgresql-client`, oder nutzen Sie ein Cross-Platform-Tool für die GUI wie z.B. pgAdmin.
 2. Kopieren Sie den Benutzernamen und das Passwort des Supabase-Admins, und loggen Sie sich mit `psql -h db.<project-id>.supabase.co -U postgres -W` ein. Achten Sie hier auf die Domainendung .co und nicht .com.
 3. Führen Sie die SQL-Statements aus `pg_dump.sql` aus, um die Tabellen zu erstellen.
5. Richten Sie den S3-Storage auf <https://supabase.com/dashboard/project/cdnbpscedvrlglygkyn/storage/buckets> ein. Erstellen Sie hierfür einen neuen Bucket, nennen Sie ihn “restaurant_assets” und richten Sie die folgenden Policies ein:
 1. Name: Allow everyone to download and list assets
 - SELECT default: USING (bucket_id = 'restaurant_assets'::text)
 2. Name: Only allow authed users to update their own assets
 - UPDATE authenticated: USING ((bucket_id = 'restaurant_assets'::text) AND ((auth.uid())::text = (storage.foldername(name))[1]))
 3. Name: Only allow authed users to insert
 - INSERT authenticated: USING ((bucket_id = 'restaurant_assets'::text) AND ((auth.uid())::text = (storage.foldername(name))[1]))
 4. Name: Only allow authed users to delete their own assets
 - DELETE authenticated: USING ((bucket_id = 'restaurant_assets'::text) AND ((auth.uid())::text = (storage.foldername(name))[1]))
6. Setzen Sie den Bucket als “Public”, und stellen Sie ein Uploadlimit ein, welches für die Headerbilder von Kategorien auf der Speisekarte sinnvoll ist. Wir empfehlen 8MB.
7. Erlauben Sie nur Bilderuploads durch das setzen der folgenden MIME-Typen: `image/jpg`, `image/jpeg`, `image/png`, `image/webp`.

2.7.1.3 Stripe.com

Stripe ist ein Zahlungsanbieter, der es uns ermöglicht, Zahlungen von Endkunden zu akzeptieren, und diese an die Gastronomen weiterzuleiten. Stripe bietet eine sehr gute Dokumentation, und ist sehr einfach zu verwenden. Außerdem ist Stripe sehr günstig und bietet das Stripe-Dashboard für unsere Gastronomen an, welche darauf ihre Umsätze anschauen können.

1. Erstellen Sie ein neues Projekt auf Stripe.com
2. Verifizieren Sie ihr Konto, indem Sie die notwendigen Dokumente hochladen, oder verwenden Sie den Testmodus ohne Verifizierung (dies ist für den Produktivbetrieb nicht empfohlen, jedoch sinnvoll in einer Testumgebung).
3. Melden Sie sich für Stripe-Connect an.
4. Navigieren Sie zu den Entwickler-Einstellungen und kopieren Sie den “Geheimschlüssel als” als STRIPE_SECRET_KEY in die .env-Datei

5. Navigieren Sie zu den Webhooks und erstellen Sie einen neuen Webhook mit der URL `https://<domain>/api/stripe/webhook`. Wählen Sie als “Event-Typen” alles mit dem Präfix “checkout.session.*” aus. Kopieren Sie den “Signing Secret” als `STRIPE_WEBHOOK_SECRET` in die .env-Datei.
6. Legen Sie die URL fest, zu welcher navigiert werden soll, wenn ein Gastronom fertig mit der Erstellung seines Stripe-Subkontos ist. Tragen Sie diese als `STRIPE_RETURN_URL` und `STRIPE_REFRESH_URL` in die .env-Datei ein.

2.7.1.4 Root-URL

Tragen Sie ihre Root-URL als `NEXT_PUBLIC_ROOT_URL` in die .env-Datei ein. Diese wird für die Generierung von Links verwendet, z.B. für die Bestellbestätigung, oder für die Weiterleitung nach dem Checkout. Tragen Sie außerdem für eine Produktiventwicklung die Umgebungsvariable `env` als `production` ein, damit weniger geloggt wird.

2.7.1.5 Node.js

Nachdem wir erfolgreich die Secrets und Drittdienste konfiguriert haben, können wir nun die Anwendung starten. Hierfür benötigen wir Node.js. Installieren Sie Node.js, z.B. mit `sudo apt install nodejs`, installieren Sie die Abhängigkeiten mit `npm ci` in dem Verzeichnis, wo die `package.json` liegt, und führen Sie die Anwendung mit `npm run dev` aus. Die Anwendung ist nun unter `http://localhost:3000` erreichbar.

Um einen Build für die Produktion zu erstellen, führen Sie `npm run build` und `npm start` aus.

2.8 Einführung der Nutzung des Produkts durch den Endnutzer

2.8.1 Registrierung und Login

Zu Demonstrationszwecken können Sie sich mit dem Testaccount `test@wertfrei.org` und dem Passwort `test123456` einloggen.

2.8.1.1 Registrierung

1. Navigieren Sie zu `https://www.gastrobit.de/login` und klicken Sie auf “Hast du noch kein Konto? Registrieren”.
2. Geben Sie Ihre gewünschte Email-Adresse und ein Passwort ein, und klicken Sie auf “Registrieren”.
3. Sie bekommen nun eine E-Mail mit einem Verifizierungslink. Klicken Sie auf diesen Link.

2.8.1.2 Login

1. Navigieren Sie zu `https://www.gastrobit.de/login` und geben Sie Ihre E-Mail-Adresse und Ihr Passwort ein.
2. Drücken Sie auf Login.

Falls Sie Ihr Passwort vergessen haben, können Sie auf “Passwort vergessen” klicken, und Sie bekommen eine E-Mail mit einem Link zum Zurücksetzen des Passworts. Oder Sie loggen sich mit der Hilfe eines magischen Links ein.

The screenshot shows a login form with a light gray background. At the top, the word "Anmelden" is displayed in a dark blue font. Below it is a text input field labeled "E-Mail Adresse" containing the email "test@wertfrei.org". Underneath is another text input field labeled "Passwort erstellen" containing a series of dots ("....."). A large dark red button at the bottom is labeled "Anmelden" in white. To the right of the button, there are three links: "Einen magischen Link per E-Mail versenden", "Passwort vergessen?", and "Hast du noch kein Konto? Registrieren".

Figure 4: Login-Form

2.8.2 Restaurantübersicht

Nachdem Sie sich eingeloggt haben, werden Sie auf die Restaurantübersicht weitergeleitet. Hier können Sie Restaurants hinzufügen, oder bereits erstellte Restaurants verwalten.

Name	Bestellungen	Umsätze	Restauranteinstellungen	Karte editieren	Zur Website
Marc's Pizzaland	Bestellungen	Umsätze	Restauraneinstellungen	Karte editieren	Zur Website
Test12345-Unternehmen	Bestellungen	Umsätze	Restauraneinstellungen	Karte editieren	Zur Website
FH SWF Mensa	Bestellungen	Umsätze	Restauraneinstellungen	Karte editieren	Zur Website
coolrestaurant123	Bestellungen	Umsätze	Restauraneinstellungen	Karte editieren	

Figure 5: Restaurantübersicht

1. Vermutlich haben Sie noch keines erstellt, also klicken Sie auf den Button “Restaurant hinzufügen”.
2. Geben Sie den Namen des Restaurante ein und wählen Sie ob Sie ein Unternehmen, ein Individuum oder eine Non-Profit sind, und klicken Sie auf “Restaurant erstellen”.
3. Anschließend werden Sie zu unserem Zahlungsdienstleister weitergeleitet. Bitte folgen Sie den Instruktionen auf dem Bildschirm, und antworten Sie wahrheitsgemäß.
4. Im Anschluss werden Sie wieder zurück zur Restaurantübersicht weitergeleitet und ihr neues Restaurant wird angezeigt.

Im Testmodus für Entwickler können Sie Testdaten verwenden, um den Prozess zu beschleunigen:
 Mobilnummer: +49 00000000
 2FA Code: 000000
 Vorname: Max
 Nachname: Mustermann
 E-Mail: test@stripe.com
 Geburtsdatum: 01.01.1990
 Privatadresse: Musterstraße 1, 58644 Musterstadt

Branche: Restaurants
 Produktbeschreibung: Pizza-Lieferservice

Währung: Euro

IBAN: DE89370400440532013000

2.8.3 Restaurantverwaltung

Ihr neues Restaurant ist noch nicht besuchbar. Um das zu ändern, klicken Sie in der Restaurantübersicht auf “Restauraneinstellungen”.

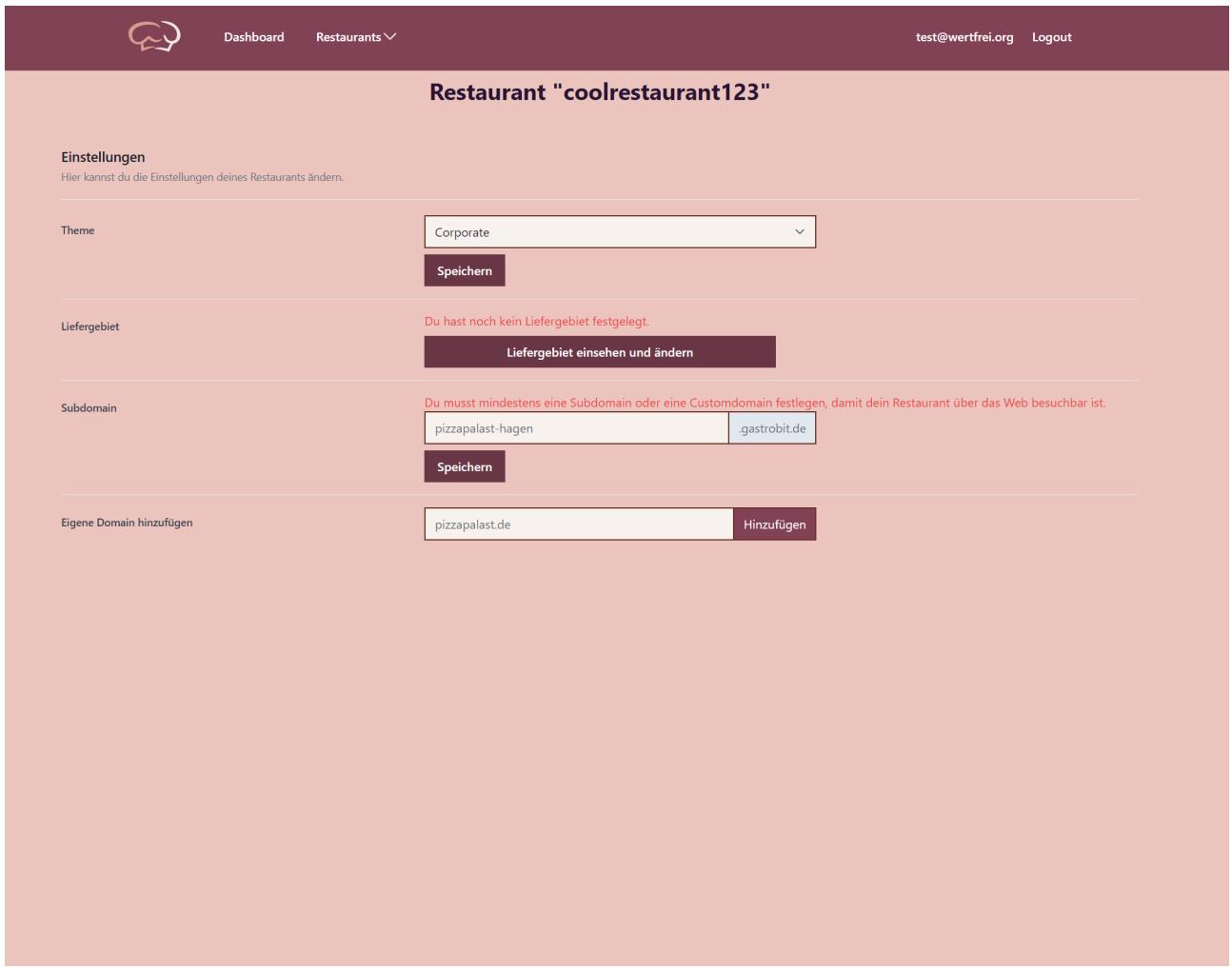


Figure 6: Restauranteinstellungen

In den Restauranteinstellungen können Sie folgende Dinge einstellen:

- eines von 21 Designs auswählen. (Die Designs sind von daisyui.com und sind unter der MIT-Lizenz veröffentlicht).
- eine Subdomain festlegen.
- eigene Domains hinzufügen.
- das Liefergebiet festlegen.

2.8.3.1 Theme

Das Theme auswählen ist sehr einfach. Klicken Sie auf ein Theme im Dropdown und klicken Sie auf “Speichern”. Das Theme wird sofort angewendet. Sie müssen jedoch erst eine Subdomain oder eine eigene Domain festlegen, und eine Speisekarte erstellen, um das Theme in voller Pracht zu sehen. Es ist sinnvoll, diese Option später neu zu besuchen, sobald wir die wichtigsten Dinge des Restaurants konfiguriert haben.

2.8.3.2 Subdomain

Hier können Sie eine Subdomain festlegen, unter der Ihr Restaurant erreichbar ist. Geben Sie einen Namen ein, und klicken Sie auf “Speichern”. Sie können nun Ihr Restaurant unter <subdomain>.gastrobit.de erreichen. Sie müssen jedoch erst eine Speisekarte erstellen, um das Restaurant zu besuchen.

2.8.3.3 Eigene Domains

Falls Sie eine eigene Domain besitzen, z.B. “marcs-pizzarestaurant-iserlohn.de”, können Sie diese hier hinzufügen. Tragen Sie die Domain in das Feld ein und drücken Sie auf “Domain hinzufügen”.

Nach wenigen Sekunden bekommen Sie ein neues Element auf Ihrem Bildschirm angezeigt, welches Ihnen sagt, dass Sie einen DNS-Eintrag hinzufügen müssen. Dieser Eintrag ist notwendig, damit wir Ihre Domain mit

unserem System verbinden können. Sie können hier zwischen “A-Record” und “CName-Record” wählen.

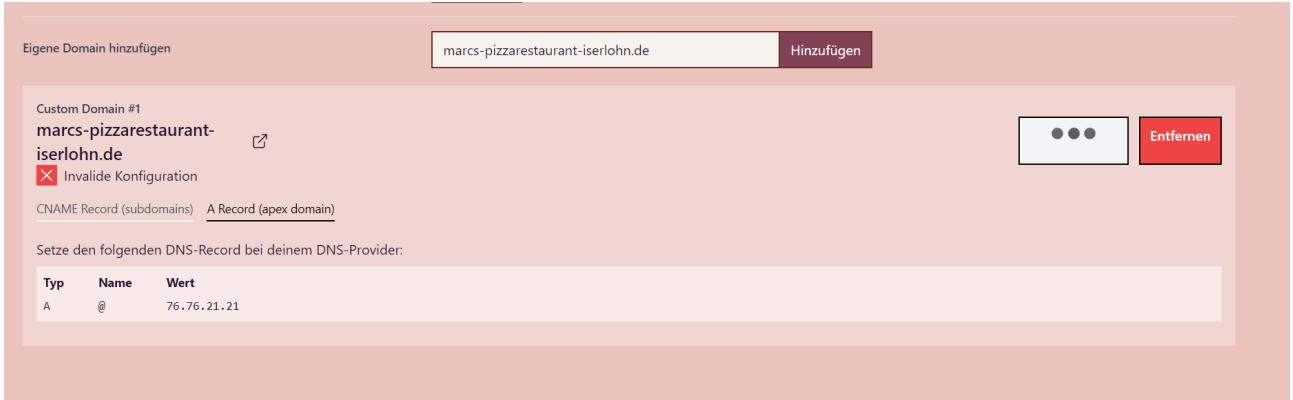


Figure 7: Eine invalide konfigurierte Domain

Falls Sie nicht wissen, welchen Eintrag Sie hinzufügen müssen, kontaktieren Sie bitte ihren Domain-Provider.

Nachdem Sie den Eintrag hinzugefügt haben wird nach wenigen Sekunden ihre Domain als “Verifiziert” angezeigt. Sie können nun ihr Restaurant unter <domain> erreichen. Sie müssen jedoch erst eine Speisekarte erstellen, um das Restaurant zu besuchen.

The screenshot shows the Cloudflare DNS records panel. At the top, there is a search bar with "Search DNS Records", a "Add filter" button, a search input field, a "Search" button, and a "Add record" button. Below the search bar, a table lists the DNS records:

Type	Name	Content	Proxy status	TTL	Actions
A	helloworld	76.76.21.21	DNS only	Auto	Edit ➔

Figure 8: Eine Domain im Panel vom Domainanbieter “Cloudflare.com”, welche korrekt konfiguriert wurde.

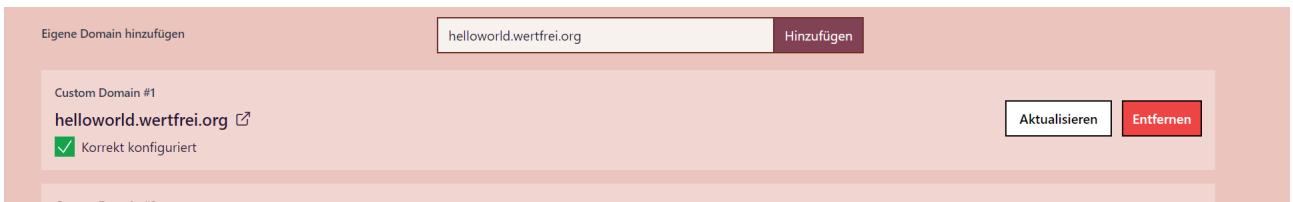


Figure 9: Eine korrekt konfigurierte Domain.

2.8.3.4 Liefergebiet

Um ein Liefergebiet festzulegen, folgen Sie diesen Instruktionen:

1. Klicken Sie auf den Button “Liefergebiet einsehen und festlegen”.
2. Sie werden aufgefordert Ihren Standort zu teilen. Lesen Sie die Meldung durch, verstehen Sie sie, und klicken Sie ggf. auf “Erlauben”.

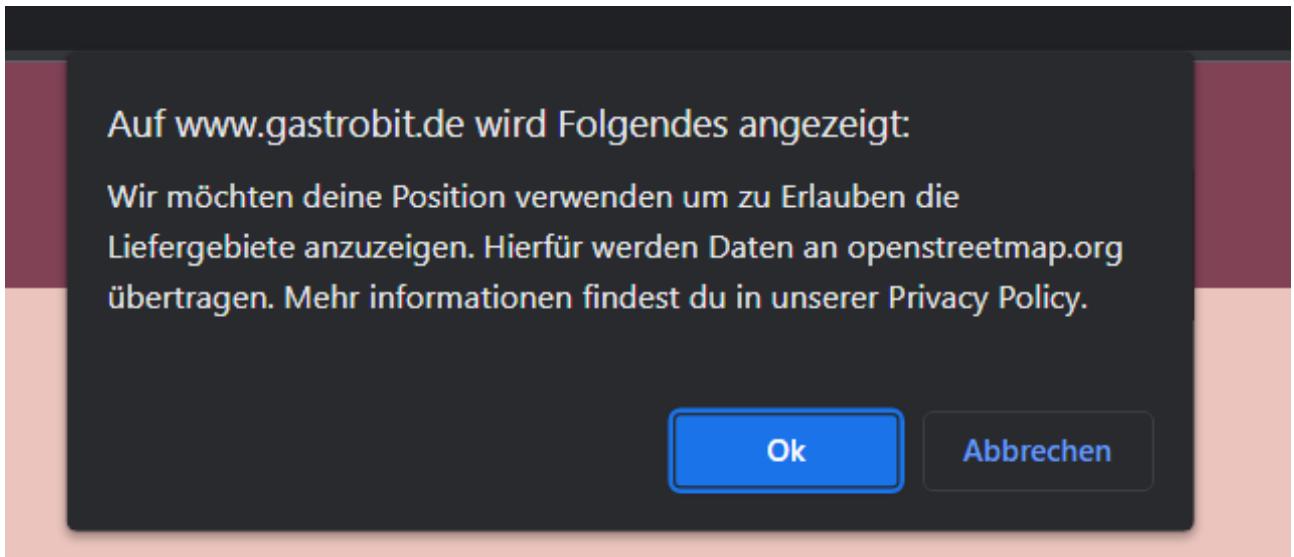


Figure 10: Standort teilen? Ja.

3. Sie werden nun auf eine Karte weitergeleitet, auf der Sie ihr Liefergebiet festlegen können. Benutzen Sie dafür das "Pen"-Werkzeug, um neue Punkte auf der Karte einzutragen. Sie können auch bereits bestehende Punkte per Drag-and-Drop verschieben. Wenn Sie einen Punkt löschen wollen, klicken Sie den Punkt an und anschließend auf "Delete".
4. Klicken Sie auf "Speichern", wenn Sie fertig sind.

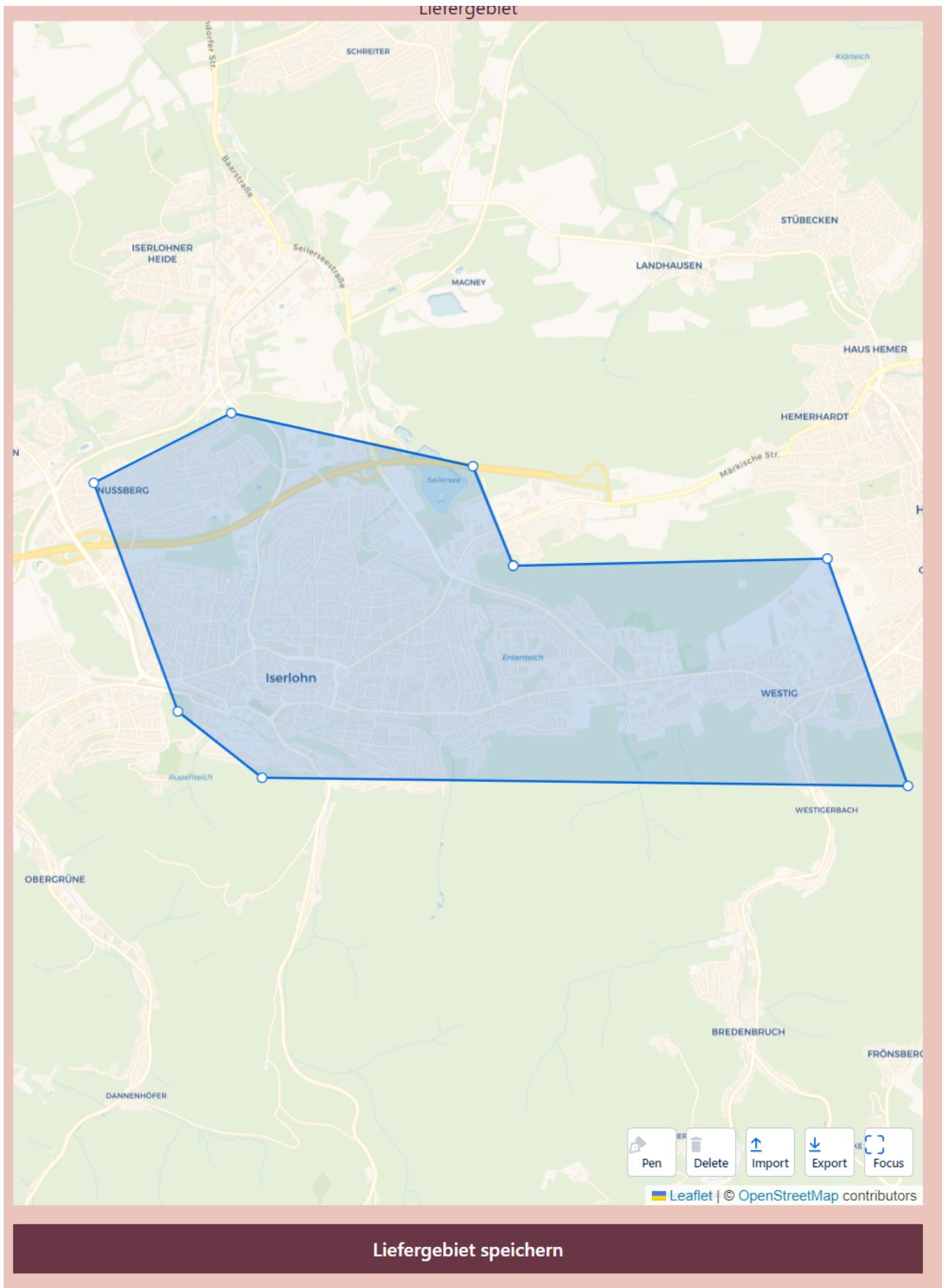


Figure 11: Polygon für Ost-Iserlohn und Hemer.

2.8.4 Speisekarte

Um eine Speisekarte zu erstellen, klicken Sie auf “Karte editieren” in der Restaurantübersicht. Anschließend landen Sie bei der Speisekartenverwaltung, mit ein paar Beispielen, die Sie löschen können.

The screenshot shows a web-based food menu management system. At the top, there's a dark header bar with a logo, 'Dashboard', 'Restaurants', and user information ('test@wertfrei.org', 'Logout'). Below the header, the main area has a light brown background. On the left, there's a section titled 'Speisekarte' featuring a photo of a salad with chicken. Below the photo is a title 'Pizza' with up/down arrows and a red '+'/ '-' button. A list of pizza options follows:

- Pizza Fungi
mit Mozzarella, Schinken und Rucola
klein (18cm): 7.5
mittel: 8.5
- Pizza Prosciutto
mit Mozzarella, Schinken und Rucola
klein (18cm): 7.5
mittel: 8.5
- Pizza Tuna
mit Mozzarella, Thunfisch und Rucola
klein (18cm): 7.5

On the right side, there's another section titled 'Presets für Extras' with a 'Preset hinzufügen +' button. Below it is a photo of a salad with chickpeas and vegetables. Below the photo is a title 'Salate' with up/down arrows and a red '+'/ '-' button. A list of salad options follows:

- Gemischter Salat
mit Eisbergsalat, Tomaten, Gurken und Mais
klein (18cm): 7.5

At the bottom of the page are two buttons: 'Neue Kategorie hinzufügen' and 'Speichern'.

Figure 12: Initiale Speisekarte

2.8.4.1 Presets für Extras

Ein Extra ist auf Extra-Attribut zu einem Gericht. Dies kann bei einer Pizza beispielsweise “Käse”, “Zwiebeln”, “Paprika”, etc. sein. Sie können hierfür Presets erstellen, um die Arbeit zu erleichtern.

Es gibt zwei Arten von Extras: **Einzelauswahl** und **Mehrfachauswahl**.

Einzelauswahl bedeutet, dass der Kunde nur eine Option auswählen kann, wie z.B. die Soße bei einer Pizza (man wählt zwischen Tomatensoße oder Hollondaise). Wenn eine Auswahl getroffen werden muss, kann es hier sinnvoll sein, den Preis für die Auswahl auf "0"€ zu setzen.

Klicken Sie auf "Extra hinzufügen" um eine neue Zeile hinzuzufügen, oder auf das Kreuz um eine Zeile zu löschen. Drücken Sie auf "Hinzufügen", um das Preset fertigzustellen.

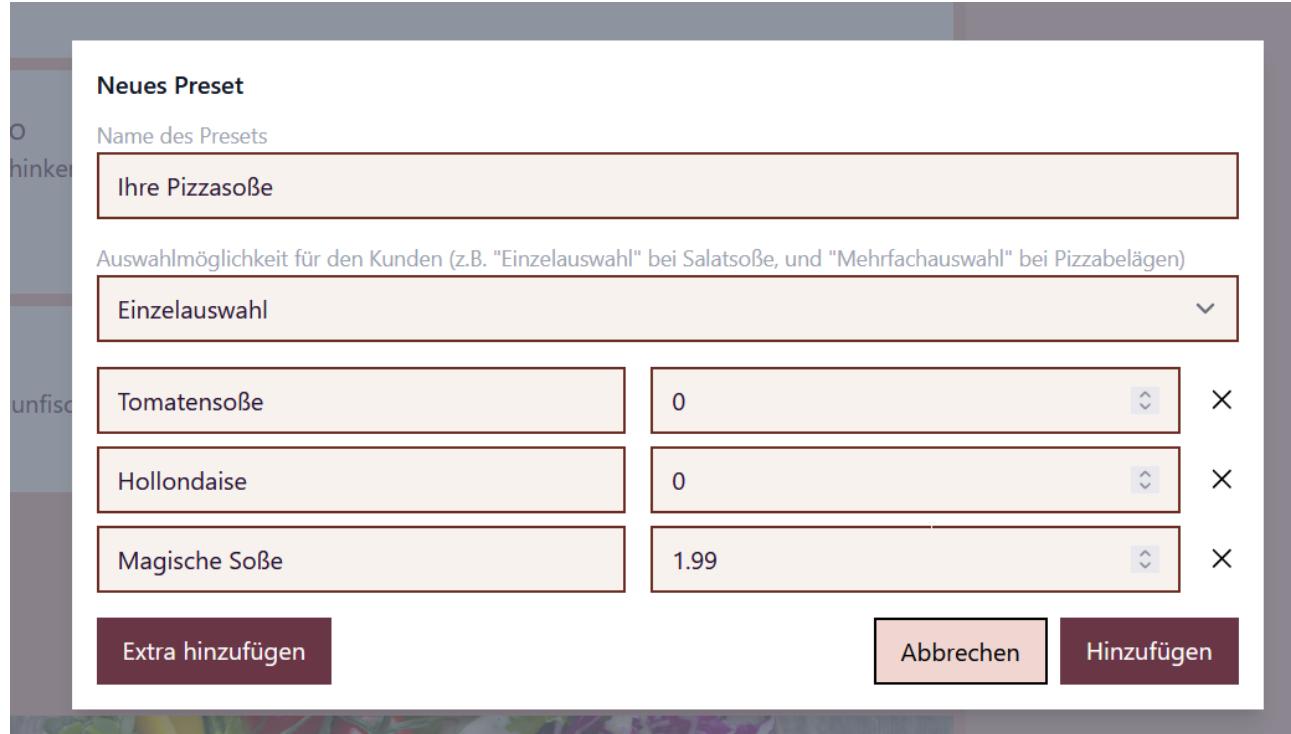


Figure 13: Einzelauswahl

Mehrfachauswahl bedeutet, dass der Kunde mehrere Optionen auswählen kann, z.B. kann der Kunde mehrere Extras wie Käse, Zwiebeln, Paprika etc. auswählen.

Das ganze funktioniert genau wie die Einzelauswahl:

Neues Preset

Name des Presets

Auswahlmöglichkeit für den Kunden (z.B. "Einzelauswahl" bei Salatsoße, und "Mehrfachauswahl" bei Pizzabelägen)

Mehrfachauswahl

Thunfisch	0.50	X
Zwiebel	1	X
Käse	0.5	X
Paprik	0.5	X

Extra hinzufügen Abbrechen Hinzufügen

Figure 14: Mehrfachauswahl

Bitte beachten Sie die Daten zu speichern durch das klicken auf “Speichern” am unteren Ende der Seite.



Figure 15: Speichern nicht vergessen.

2.8.4.2 Kategorien

Sie können eine Kategorie einfach durch das drücken auf “Neue Kategorie hinzufügen” am unteren Ende der

Seite anklicken. Sie werden darauf aufgefordert, einen Namen für die Kategorie festzulegen.

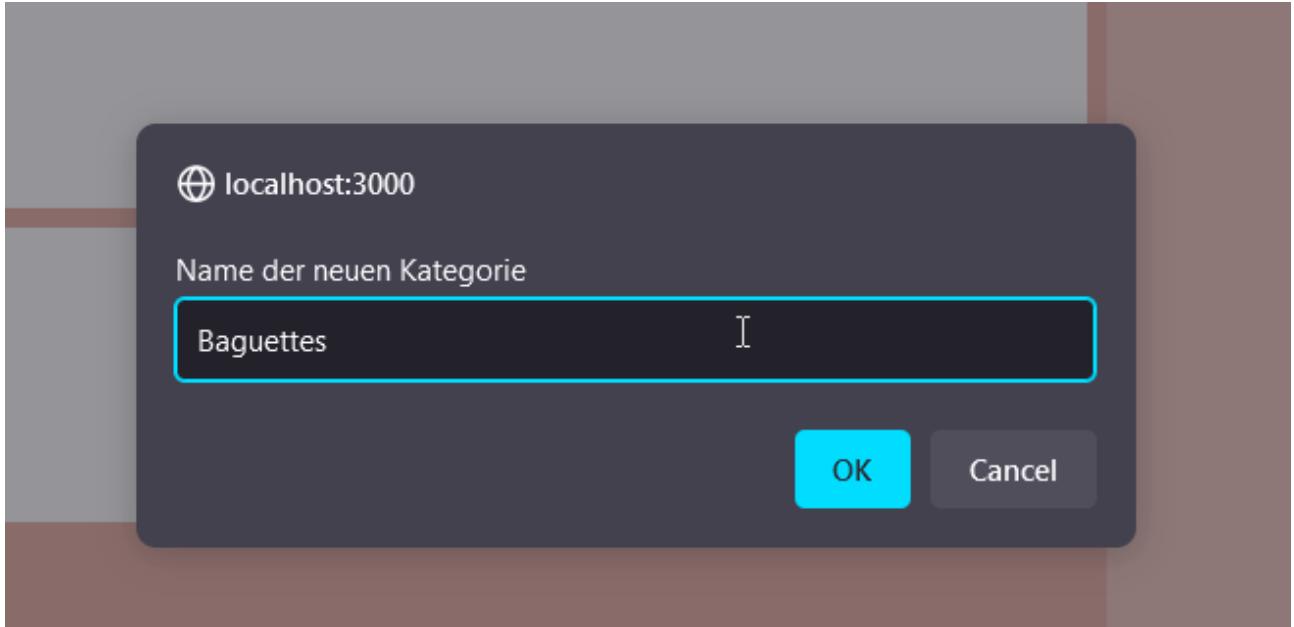


Figure 16: Geben Sie den Namen der neuen Kategorie an.

Anschließend können Sie einen neuen Banner per Drag-and-Drop auf den Platzhalter, oder durch Anklicken setzen.



Figure 17: Neues Banner setzen.

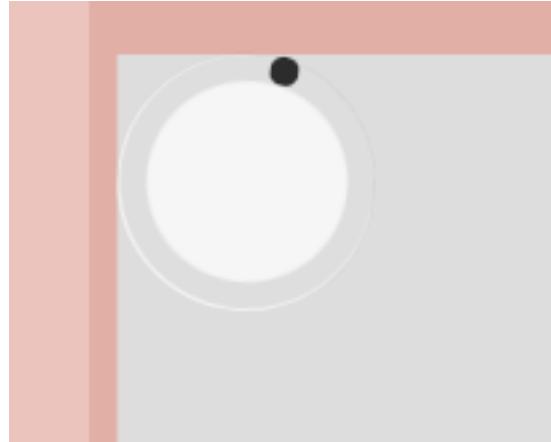


Figure 18: Nicht die Seite verlassen, während der Ladeindikator bei dem neuen Banner angezeigt wird. In dieser Zeit wird das Bild hochgeladen, was je nach Internetverbindung ein wenig dauern kann.

Sie können die Kategorie durch Klicken auf den Namen wieder umbenennen, oder durch das Klicken auf das Kreuz löschen.

Die Pfeile erlauben es Ihnen, die Reihenfolge der Kategorien zu ändern.

Das Kreuz neben dem Namen der Kategorie erlaubt es Ihnen, die Kategorie zu löschen.

Bitte klicken Sie auf “Speichern”, um die Änderungen zu speichern.

2.8.4.3 Gerichte

Fügen Sie ein neues Gericht zu einer Kategorie hinzu, indem Sie auf das Plussymbol klicken.

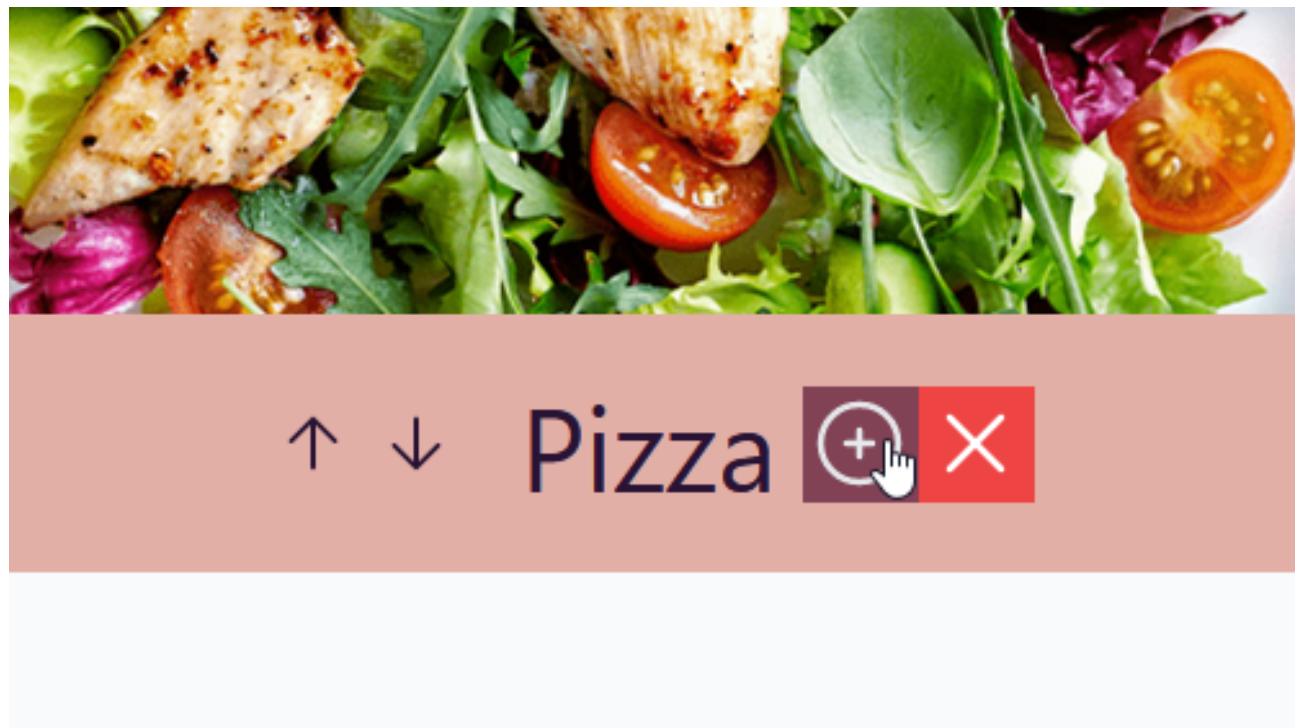


Figure 19: Plussymbol anklicken.

Anschließend können Sie das neu hinzugefügte Gericht bearbeiten, indem Sie auf den Strift oben rechts in der Ecke klicken. *Übrigens, mit den zwei Strichen daneben können Sie das Gericht in der Reihenfolge der Kategorie verschieben. Einfach draufklicken, die Maustaste gedrückt halten, und das Gericht an die gewünschte Stelle ziehen. Das Kreuz daneben löscht logischerweise das Gericht.*

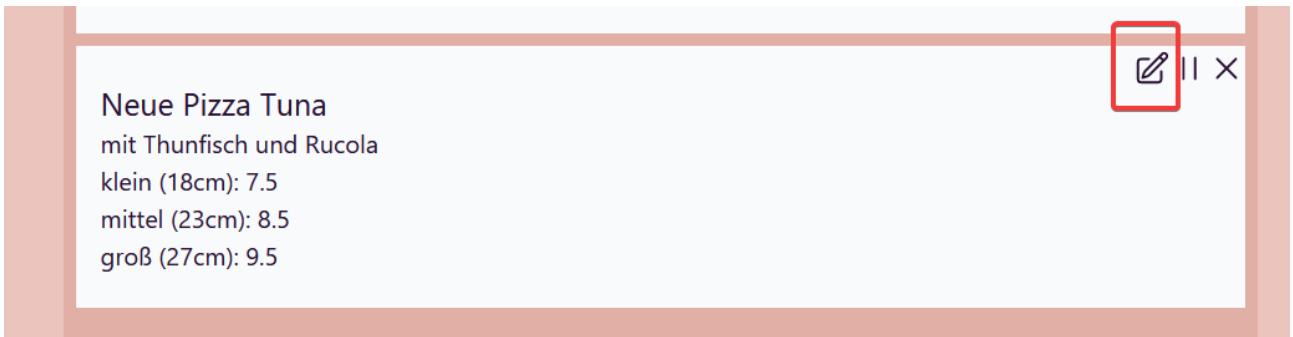


Figure 20: Gericht bearbeiten.

Anschließend kommt das folgende Menü zum Vorschein:

Neue Pizza Tuna



Hier kannst du die Daten des Gerichts ändern. Oben ist der ursprüngliche Name des Gerichtes, damit du nicht ausversehen ein falsches Gericht editierst.

Überschrift (z.B. "Pizza Prosciutto")

Pizza Magie

Unterschrift

mit Magie und Hass

1

Preise

klein (18cm)

7.5



mittel (23cm)

8.5



groß (27cm)

9.5



Extra-Attribute

Ihre Pizzaextras



Hinzufügen

Figure 21: Editieren des Gerichtes

Sie können die Überschrift und Unterschrift hier bearbeiten.

Hier können Sie Preise für verschiedene Varianten des Gerichtes (z.B. Groß, Mittel, Klein bei Pizzen) festlegen. Wenn es nur eine Variante gibt, ist es ratsam, die Variante einfach "Standard" zu nennen, und die restlichen Varianten zu löschen.

Außerdem können Sie hier die Extra-Attribute aus ihren Presets hinzufügen. Einfach in der Combobox auswählen und auf "Hinzufügen" drücken. Sie können die Extras auch wieder löschen, indem Sie auf das Kreuz neben dem Namen des Extras klicken.

Extra-Attribute

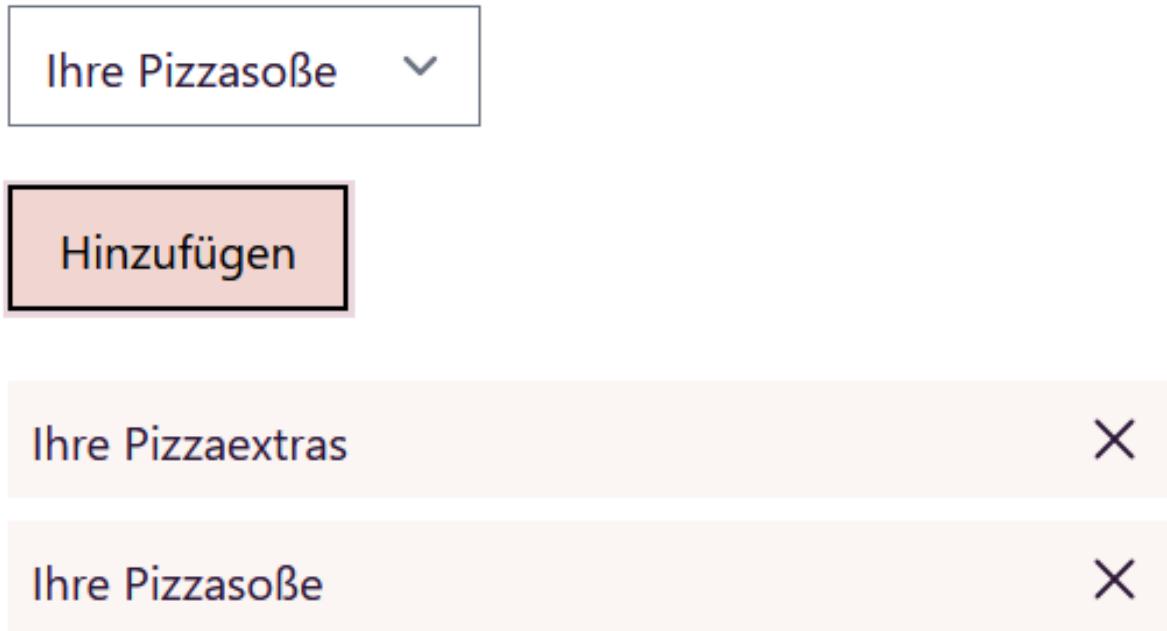


Figure 22: Extras

Wenn Sie fertig sind, klicken Sie auf "Ändern", und anschließend auf "Speichern" am Ende der Seite.

2.8.5 Bestellen

2.8.5.1 Gerichtsauswahl

Gehen Sie auf die Seite Ihres eingerichteten Restaurants. Benutzen Sie dafür Ihre Custom-Domain oder eine Subdomain. Wir haben eine Demoseite auf <https://marcpizzaland.gastrobit.de/> vorbereitet.

Speisekarte



Burger

Cheeseburger
mit Käse

AB 7.5 €

Warenkorb

Cheeseburger - 7.5€
Standard
Ihre Burgersoße: Ketchup

X

BESTELLEN FÜR 7.5€



Salate

Gemischter Salat
mit Eisbergsalat, Tomaten, Gurken und Mais

AB 7.5 €

Eisbergsalat
mit Zwiebel

AB 7.5 €



Pizza

Pizza Prosciutto
mit Mozzarella, Schinken und Rucola

AB 7.5 €

Pizza Tuna
mit Mozzarella, Thunfisch und Rucola

AB 7.5 €

Pizza Salami
mit Mozzarella, Schinken und Rucola

AB 9.5 €

Figure 23: Ein Demorestaurant.

Die Bedienung ist selbsterklärend, schließlich sollen Ihre Kunden sich auch zurechtfinden. Sie können Gerichte in den Warenkorb legen, Extras auswählen (sofern konfiguriert), und anschließend auf "Bestellen für X€" klicken. Der Preis auf dem Button aktualisiert automatisch.

Pizza Tuna

mit Mozzarella, Thunfisch und Rucola

Variante

mittel (23cm) - 10 €

Attribute

Ihre Pizzaaextras

Thunfisch +0.5€



Mozzarella +0.5€

Tomate +0.5€

Gyros +0.5€

Ihre Pizzasoße

Tomate +0€

SCHLIESSEN

FÜR 11€ ZUM WARENKORB HINZUFÜGEN

Figure 24: Extras auswählen.

Warenkorb

Cheeseburger - 7.5€ X
Standard
Ihre Burgersoße: Ketchup

Pizza Tuna - 11€ X
Mittel (23cm)
Ihre Pizzasoße: Tomate
Ihre Pizzaextras: Thunfisch, Tomate

BESTELLEN FÜR 18.5€

Figure 25: Warenkorb

2.8.5.1.1 Lieferadresse

Die nun anzugebende Lieferadresse muss im vorher definierten Liefergebietpolygon liegen. Für den sogenannten "Lookup" von der Adresse verwenden wir OpenStreetMap. Dies ist ein freies Projekt, welches von Freiwilligen gepflegt wird. Es ist daher wichtig in der Datenschutzerklärung darauf hinzuweisen, dass die Daten an OpenStreetMap übertragen werden. Die Einwilligung muss dafür erteilt werden, weshalb wir auf die Checkbox nicht verzichten dürfen.

Außerdem ist es wichtig, dass die Daten nicht für andere Zwecke verwendet werden dürfen, als für die Bestellung. Dies ist ebenfalls in der Datenschutzerklärung zu erwähnen. Wir übernehmen das für Sie.

Einkaufswagen

Cheeseburger - 7.5€ Standard Ihre Burgersoße: Ketchup	X
Pizza Tuna - 11€ Mittel (23cm) Ihre Pizzasoße: Tomate Ihre Pizzaextras: Thunfisch, Tomate	X

Lieferadresse

Name auf der Klingel
Marc Richts

Straße und Hausnummer
Harkortstraße 3

Postleitzahl
57673

Ort
Hemer

Email (für Rechnung und Bestellbestätigung)
marc@wertfrei.org

Handynummer (optional)

Ich stimme der Verarbeitung meiner Daten gemäß der [Datenschutzerklärung](#) zu.

KOSTENPFLICHTIG BESTELLEN FÜR 18.5€

WARENKORB LEEREN

Figure 26: Checkout

Sollte die Adresse nicht im Liefergebiet liegen, bekommt der Nutzer einen Hinweis darauf.

Anschließend wird der Nutzer zur Zahlungsseite weitergeleitet.

Hier können Sie, falls Sie Stripe im Demomodus verwenden, folgende Testdaten verwenden:

Kartennummer: 4242 4242 4242 4242

Ablaufdatum: Ein Datum in der Zukunft

CVC: Drei Ziffern Ihrer Wahl

Name: Ein Name Ihrer Wahl

Land: Ein Wunschland

The screenshot shows a payment interface for a restaurant named "Gastrobitt". At the top left, there's a logo and the text "Gastrobitt TEST MODE". Below it, a summary of the order: "Cheeseburger and 1 more" with a total of "€18.50". The breakdown shows a "Cheeseburger standard "Ketchup"" at "€7.50" and a "Pizza Tuna mittel (23cm) "Tomate, Thunfisch, Tomate"" at "€11.00". On the right side, there's a "Contact information" section with an email field containing "marc@wertfrei.org". Below that is a "Payment method" section with four options: "Card", "SOFORT", "Klarna", and "giropay". Further down are sections for "Card information" (with a placeholder card number "4242 4242 4242 4242" and expiration date "01 / 33" and CVV "111"), "Name on card" ("Stripe Test Card"), "Country or region" ("Finland"), and a "Securely save my information for 1-click checkout" section with a phone number input ("+ 041 2345678") and an "Optional" link. At the bottom is a large green button with a white checkmark icon.

Gastrobitt TEST MODE

Cheeseburger and 1 more

€18.50

Cheeseburger standard "Ketchup"	€7.50
Pizza Tuna mittel (23cm) "Tomate, Thunfisch, Tomate"	€11.00

Contact information

Email: marc@wertfrei.org

Payment method

Card SOFORT Klarna giropay

Card information

4242 4242 4242 4242 VISA

01 / 33 111

Name on card

Stripe Test Card

Country or region

Finland

Securely save my information for 1-click checkout

Enter your phone number to create a Link account and pay faster on Gastrobitt and everywhere Link is accepted.

+ 041 2345678 Optional

link • More info

Powered by stripe | [Terms](#) [Privacy](#)

Figure 27: Zahlen, bitte.

2.8.6 Bestellungen abarbeiten

In der Restaurantübersicht können Sie auf "Bestellungen" klicken, um die Bestellungen zu verwalten.

Diese Ansicht ist Live. Das bedeutet, dass sobald eine neue Bestellung eingeht, diese Seite automatisch aktualisiert wird. Dies wird mit Websockets realisiert, und ist sehr performant. Sie können die Seite also einfach offen lassen, und Sie werden benachrichtigt, sobald eine neue Bestellung eingeht.

Marc's Pizzaland					
Orders					
Bestelldatum	Adresse		Lieferstatus	Email	Bezahlstatus
7/18/2023, 11:32:39 PM	Marc Richts, Harkortstraße 3, 57673 Hemer	Zeige Bestellinformationen	Offen	marc@wertfrei.org	pending
7/18/2023, 8:34:38 PM	Marc Richts, Harkortstraße 3, 58675 Iserlohn	Zeige Bestellinformationen	Offen	marc@wertfrei.org	pending
7/18/2023, 4:32:04 PM	Marc Richts, Harkortstraße 3, 57673 Hemer	Zeige Bestellinformationen	In Bearbeitung	marc@wertfrei.org	paid
7/18/2023, 4:19:03 PM	Marc Richts, Harkortstraße 3, 57673 HEmer	Zeige Bestellinformationen	Offen	marc@wertfrei.org	pending
7/18/2023, 4:16:59 PM	Marc Richts, Harkortstraße 3, 57673 Hemer	Zeige Bestellinformationen	Ausgeliefert	marc@wertfrei.org	paid
7/18/2023, 4:15:11 PM	Marc Richts, Harkortstraße 3, 57673 Hemer	Zeige Bestellinformationen	Abgelehnt	marc@wertfrei.org	pending
7/18/2023, 4:13:04 PM	Marc Richts, Harkortstraße 3, 57673 Hemer	Zeige Bestellinformationen	Abgelehnt	marc@wertfrei.org	pending
7/18/2023, 4:08:03 PM	Marc Richts, Harkortstraße 3, 57673 Hemer +4915734031337	Zeige Bestellinformationen	Offen	marc@wertfrei.org	pending

Figure 28: Bestellungsübersicht

Wenn Sie auf "Zeige Bestellinformationen" klicken, bekommen Sie die Orderdetails angezeigt:

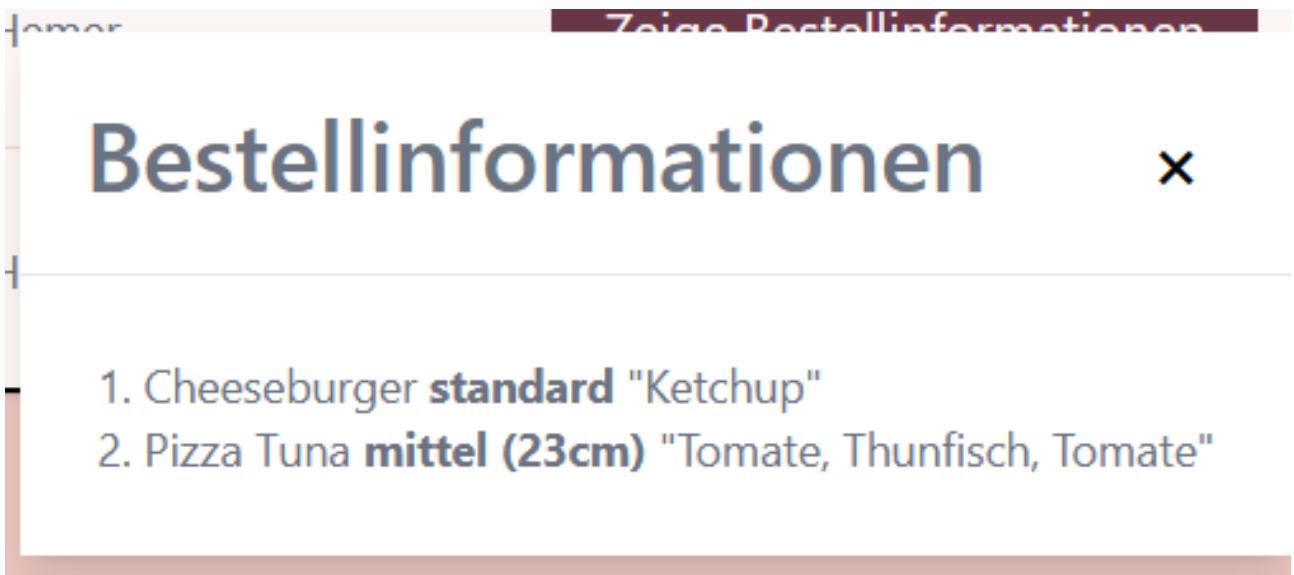


Figure 29: Bestellinformationen, die jemand in der Küche hoffentlich umsetzt.

Hier können Sie außerdem den Lieferstatus verändern, und einsehen, ob die Lieferung bereits bezahlt wurde. Mit den meisten Zahlungsmöglichkeiten wie Kreditkarte geht das innerhalb weniger Sekunden. Wir empfehlen, keine Bestellungen zu bearbeiten, bevor die Zahlung nicht abgeschlossen ist.

An dieser Stelle hatten wir die Designentscheidung getroffen, auch nicht-bezahlte Bestellungen anzuzeigen, einfach aus dem Grund, dass wir der Meinung sind, dass die Daten dem Gastronomen gehören, und er die Verantwortung dafür übernehmen soll.

2.8.7 Lieferstatus

Der Lieferstatus kann folgende Werte annehmen: - Offen - In Bearbeitung - Ausgeliefert - Abgelehnt

Die ersten drei Status dienen nur der Verwaltung und Übersicht für Sie. Abgelehnt bedeutet jedoch, dass Sie die Bestellung nicht ausliefern können (z.B. weil Sie bereits so viel zu tun haben, oder nicht genug Zutanten haben).

Sollten Sie diese Option auswählen, wird ein automatischer Refund an den Kunden getriggert, und Sie zahlen 35 Cent Gebühren an Stripe (sofern Sie auf Stripe wirklich Live sind, und nicht den Testmodus verwenden).

2.9 Quellen

Falls Sie weitere Fragen haben, kontaktieren Sie unseren Support unter richts.marc-alexander@fh-swf.de.

Quellen und verwendete Libraries, sofern nicht anders markiert:

```
"@dnd-kit/core": "^6.0.8",
"@dnd-kit/modifiers": "^6.0.1",
"@dnd-kit/sortable": "^7.0.2",
"@freenow/react-polygon-editor": "^2.0.1",
"@headlessui/react": "^1.7.13",
"@heroicons/react": "^2.0.17",
"@prisma/client": "4.12.0",
"@supabase/auth-helpers-nextjs": "^0.6.1",
"@supabase/auth-helpers-react": "^0.3.1",
"@supabase/auth-ui-react": "^0.3.5",
"@supabase/auth-ui-shared": "^0.1.3",
"@supabase/supabase-js": "^2.15.0",
"@types/node": "18.15.11",
"@types/react": "18.0.31",
"@types/react-dom": "18.0.11",
"daisyui": "^3.1.0",
"eslint": "8.37.0",
"leaflet": "^1.9.4",
"lodash-es": "^4.17.21",
"micro-cors": "^0.1.1",
"next": "^13.4.1",
"node-uuid": "^1.4.8",
"nominatim-browser": "^2.1.0",
"nominatim-client": "^3.2.1",
"querystring": "^0.2.1",
"react": "^18.2.0",
"react-dom": "^18.2.0",
"react-dropzone": "^14.2.3",
"react-geolocated": "^4.1.1",
"react-hook-form": "^7.43.9",
"react-hot-toast": "^2.4.1",
"react-leaflet": "^4.2.1",
"react-spinners": "^0.13.8",
"redux": "^4.2.1",
"stripe": "^11.18.0",
"styled-components": "^6.0.4",
"swr": "^2.1.5",
"typescript": "5.0.3",
"zod": "^3.21.4",
"zustand": "^4.3.8"
"@tailwindcss/aspect-ratio": "^0.4.2",
"@tailwindcss/forms": "^0.5.3",
"@types/leaflet": "^1.9.3",
"@types/lodash-es": "^4.17.8",
"@types/lodash.isequal": "^4.5.6",
"@types/micro-cors": "^0.1.3",
"autoprefixer": "^10.4.14",
"eslint-config-next": "^13.4.1",
"postcss": "^8.4.21",
"supabase": "^1.50.2",
"tailwindcss": "^3.3.1"
```

Bilder für die Präsentation wurden von Unsplash bezogen. Es gilt die Lizenz dass die Bilder nichtkommerziell

frei verwendet werden dürfen. Mehr Informationen dazu unter <https://unsplash.com/license>.