Dokumentation für Lieferservice von **elegia**

Team:

Mykhailo Rodchenkov

Nikita Ivachshenko

Thi Ngoc Thi Nguyen

Siti Rizki Adhaila Ilyas

Trainer:

Michael Höding

Wintersemester 2018/19, Wirtschaftsinformatik

Inhaltsverzeichnis

Executive Summary

“Finde. Kaufe. Genieße” - elegia´s Mantra

Im Rahmen der Veranstaltung Frameworks haben wir das erste Prototyp eines Lieferservices entwickelt. Es handelt sich um ein Vermittlerplattform wo die KMU das Lebensmittel für das Büro und für die Mitarbeiter besorgen können (z.B. Getränke, Snacks oder Obste). Auf der anderen Seite sind die Anbieter, die sich auf der Webseite anmelden und ihre Waren anbieten können. Momentan sind 4 Releases unter Github (siehe Teil 7) zu finden, die Version 2.0 gilt als erstes funktionsfähiges Prototyp und die nächste Version 2.1 ist zusätzlich mit der deutschen Übersetzung ergänzt worden. Es gibt noch weiße Flecken, die zu implementieren sind, eine und die wichtigste davon ist das Bezahlen von Waren. Bis v.2.1 bekommt die Bestellung den entsprechenden “Bezahlt” Status dann. wenn man auf “Bezahlen” Button klickt. An dieser Stellen würde man eine Service einbinden, welches das Prozess der Bezahlung steuert.

Die geografische Abdeckung der Anwendung umfasst bis jetzt ganz Deuschland. Die Datenbank enthält ca. 14 Tausend deutschen Städten mit PLZ und Bundesländer. Ab v.2.1 unterstützt die Anwendung zwei Sprachen - deutsch und englisch.

Es wurde ein komfortables Mechanismus für die Anbieter entwickelt, die dabei hilft, die Statusänderung der Bestellung vorzunehmen und zu steuern, damit keine Bestellung verloren geht und vergessen wird. Andererseits können die Kunden einfach die richtigen Waren bei den passenden Anbietern finden. Schönes Design des Shops nach Usability Regeln und angenehme aufregende Farben neigen dazu, die Kaufentscheidung möglichst schnell zu treffen und den Kauf abzuschließen. Deswegen haben wir einen großen Wert auf UI- und UX-Gestaltung gelegt.

Jeder Nutzer, egal ob Kunde oder Anbieter, hat eine eigene Profilseite, wo man nähere Information und Foto anlegen kann. Die Anbieter haben darüber hinaus noch weitere Funktionalitäten, wie die Ware erstellen, Bestellungen und Benachrichtigungen abchecken uws. Die Anbieter werden sofort benachrichtigt, wenn eine Bestellung kommt und die Kunden werden in Kenntnis gesetzt, wenn der Anbieter die Bestellung empfangen und in Bearbeitung genommen hat.

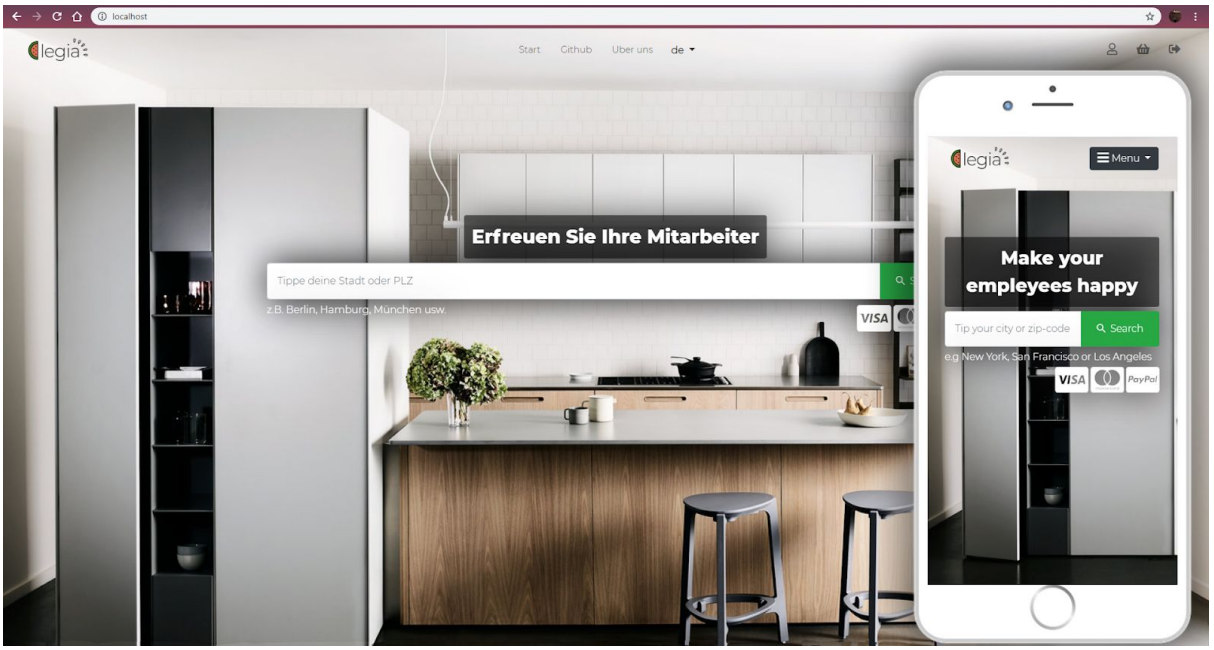
Die Rollen sind stark abgetrennt, sodass die Kunden keine Waren anbieten können und die Anbieter keine Waren erwerben können. Diese Eigenschaft wird noch im Team diskutiert, ob es nicht besser wäre, die gleichen Möglichkeiten an beide Rollen zu vergeben.

Architekturkonzept und Technologien

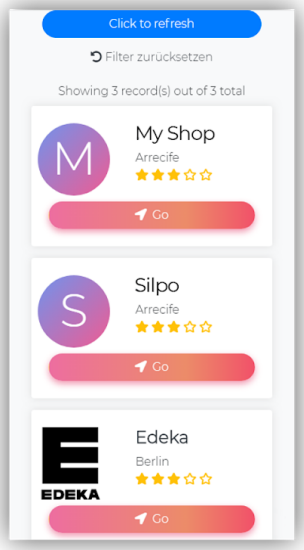
Die Entwicklungsphase wurde auf mehrere gleichmäßige Masken (OOD Design + OOD Logik) aufgeteilt, wobei jede Maske exakt einer View entspricht. Diese Menge von Masken war jedem Entwickler zur freien Wahl bereitgestellt, so dass jeder Teammitglied die Maske nach seinem Wunsch aussuchen und in Bearbeitung nehmen könnte. Jedem Entwickler wurde lokal die Entwicklungsumgebung eingerichtet. Sie enthielt lokalen Apache-Server mit PHP und MySQL und Version Control System (Git). Alle Änderungen wurden direkt in Master-Branch eingecheckt. Nach Fertigstellung und Bug-Prüfen der Maske war sie in remote Git-Repository mit dem entsprechenden Kommentar einzuchecken. Außerdem haben wir zusätzliche Batch-Skripte erstellt, die das Backup und Update der Datenbank erleichtert haben. Die Backups von DB wurden durchschnittlich ein mal pro Woche per manuelles Ausführen des Scriptes durchgeführt und von anderen Entwicklern per Script lokal aktualisiert. Im Dezember haben wir die erste Version auf **fbwframework** hochgeladen und ein mal pro Woche händisch `$ git pull origin` ausgeführt, um fbwframework auf dem aktuellen Stand zu halten.
Für das Backend haben wir CakePHP (3.6.14) benutzt. Die ersten MVC-Einheiten wurden mit cake bake generiert, die anderen manuell erstellt, wobei es sich später herausgestellt hat, dass weitere Controller überflüssig waren, und am besten zu schon generierten Controllern passten. Für das Design haben wir Bootstrap (css und js für Tooltip) ve rwendet. Für die Icons wurde das kostenlose Paket von Fontawesome verwendet. Es wurde entschieden, JavaScript zu vermeiden und nur in sehr dringenden Fällen zu verwenden, um den Komplexitätsgrad niedrig halten und möglichst viel CakePHP-Potenzial ausnutzen zu können.

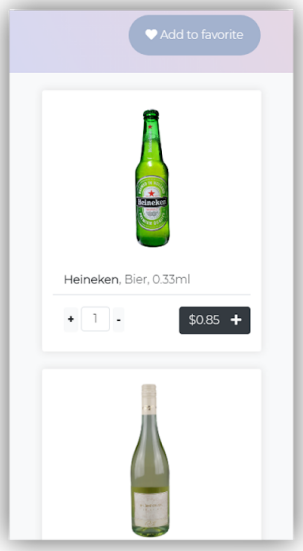
UI/UX und Responsive Design

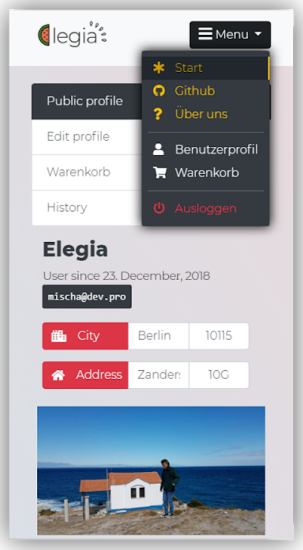
Wie oben erwähnt wurde, haben wir ausschließlich Bootstrap für das Styling verwendet, um möglichst getestete und browserübergreifende CSS-Regeln zu nutzen. Die Gestaltung von einzelnen Widgets und Elementen wurde nach Usability Regeln konstruiert. Die Buttons und die interaktiven Elemente sind nach Material Design von Google erarbeitet und entwickelt. Wir haben mit Hilfe von Card Sorting (auf jeder Karte ist eine Maske dargestellt) die Klickpfade ausgearbeitet, sodass es sich einfach lässt, durch die Webseite zu navigieren und die benötigte Seite maximal mit drei Klicks zu finden (beispielsweise **eine bestimmte Ware bearbeiten** - “Mein Shop” → “Warenliste” → “Bearbeiten”). Die Farben, die das GUI aufzeichnet, wurden gemeinsam im Team ausgewählt und eine entsprechende Farbpalette erstellt. Die Farben wurde mit der Gedanke erarbeitet, dass sie ein angenehmes Gefühl weckt damit man sich beim Einkaufen wohl fühlt und sich später beim Umschauen der Küche an **elegia** erinnert.



Das responsive Design wurde auch mit Bootstrap entwickelt und mithilfe von Browser Erweiterung für unterschiedliche Endgeräte (inkl. die neuesten mobilen Endgeräte) getestet.



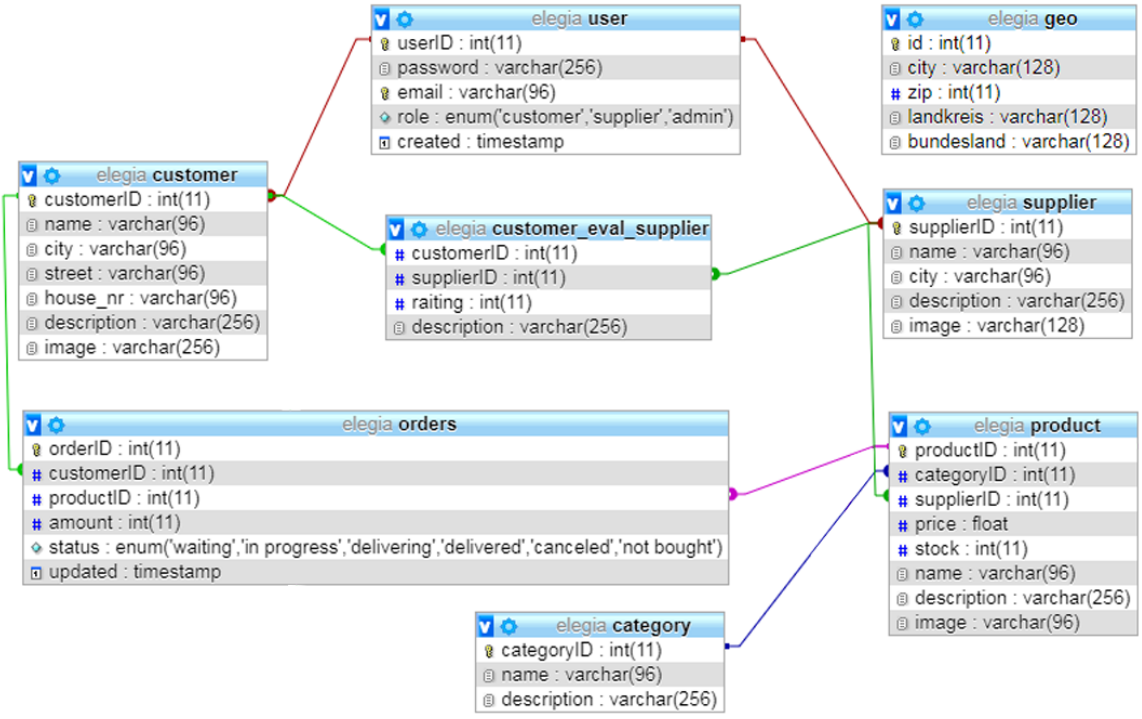




Beim Verkleinern der Seite oder bei der Nutzung von kleinen Endgeräten verstecken sich die Header-Elemente und sammeln sich in einem Dropdown-Menü, welches sich mit einem klick aufklappt. Einige Elementen verstecken sich an Geräten mit kleiner Auflösung komplett und lassen sich nicht zeigen, zum Beispiel Shopinformation in Suchansicht (/search). Aber keine Information geht verloren, sonder lässt sich auf der anderen Seite finden, z.B. ober erwähnte Shopinformation findet man auf der Seite der Anbieter.

Datenbank

Das geplante Schema der Datenbank unterscheidet sich ein bisschen von der tatsächlichen.



Wir haben zeitlich nicht geschafft, einige Funktionalitäten zu implementieren, die in der ursprünglichen Datenbank abgebildet waren.

How to use

Nicht angemeldete Benutzer können sich nur die Start-, Login- und Registrierungsseite anschauen. Deswegen muss man zuerst das Profil anlegen, dafür geht man auf **/signup**. Auf der Seite trägt man erste persönliche Information ein. Darüber hinaus muss man schon auf diesem Schritt die Rolle auswählen, zur Wahl stehen zwei Rollen: **Kunde und Lieferant**. Nachdem die Registrierung erfolgreich stattgefunden hat und die Success-Nachricht in oberen Teil der Seite erschien, kann man sich unter **/login** anmelden. Danach wird man zurück zur Startseite weitergeleitet, im Header sind nun 3 zusätzliche Links mit Icons verfügbar. Beim Eintippen der Stadt oder PLZ im Suchfeld oder einfach beim Klicken auf Suchbutton landet man auf die Seite mit Ergebnissen von eingetippten Kriterien, oder wenn man nichts eingegeben hat, werden alle Lieferanten angezeigt. Da sucht man einen Lieferanten aus und geht zu seinem Shop, wo man mit einem Klick die Waren in den Warenkorb einfügt. Bei

Klicken auf Warenkorb-Icon im Header landet man auf seinen Warenkorb und kann nun den Kauf vornehmen. Den Status der Bestellung ist in Bestellliste zu sehen.
Wenn an sich als Lieferant angemeldet hat, erscheint in seinem Shop unter **/view:id** zusätzliches Panel mit weiteren Funktionalitäten, wie z.B. Ware erstellen, Bestellungen, Benachrichtigungen, Shopansicht und Profil bearbeiten. Sobald ein Kunde eine Ware bestellt, bekommt der Lieferant ein Benachrichtigung. Diese ist unter dem Tab Benachrichtigungen oder im Header am Benachrichtigung-Icon zu sehen. Nach der Bestätigung des Eingangs der Bestellung landen die Nachrichten in Bestellungen Tab. Unter diesem Tab kann man den Status weiter ändern oder komplett ablehnen. Bei der Statusänderung aktualisiert sich das Datum, was dem Kunde auch sichtbar ist. Die Spracheinstellungen befinden sich im mittleren Teil des Headers, momentan stehen zwei Sprachen zur Verfügungen.
Am ende kann man sich ausloggen, indem man auf Ausloggen-Icon im Header klickt oder auf **/logout** geht.

Installation

$ git clone https://github.com/rodchenk/elegia.git
$ cp elegia-web/config/app.default.php elegia-web/config/app.php | nano
Die Datenbankverbindung entsprechend anpassen (Password, Benutzer- und DB-Name). Danach die Datenbank aus db_backup/scheme/backup_***.sql importieren. Danach folgendes ausführen um die Applikation zu installieren:
$ cd elegia-web | composer update

Links und Anmeldedaten

Rewe:	info@rewe.com	rewe
Lidl:	info@lidl.com	lidl
Edeka:	info@edeka.de	edeka
User:	user@elegia.com	user

Fontawesone:	https://fontawesome.com/
Material Design:	https://material.io/
Bootstrap:	https://getbootstrap.com/docs/4.0/
CakePHP:	https://book.cakephp.org/3.0/
Github-Repository:	https://github.com/rodchenk/elegia