

Einkaufsservice

In der derzeitigen Situation sollen ältere Personen Einkäufe durch jüngere Personen übernehmen lassen. Wir wollen für dieses Szenario eine einfache Web-basierte Applikation erstellen. Hilfesuchende, ältere Personen soll es ermöglicht werden, dass Sie eine „Einkaufsliste“ online stellen – dies soll natürlich möglichst einfach möglich sein (Überlegungen zu UX!). Angemeldete Freiwillige sollen einfach bekannt geben können, dass Sie einen gewissen Einkauf übernehmen möchten.

1 Funktionale Anforderungen

1.1 Einkaufsliste

- Besitzt Ersteller, Datum bis wann der Einkauf erledigt werden soll, Artikel die gekauft werden sollen
- Artikel besitzen Bezeichnung, Menge, sowie einen Höchstpreis der gezahlt werden soll
- Pro Einkaufsliste können beliebig viele Artikel angelegt werden.
- BenutzerInnen müssen sich anmelden, damit die Anwendung genutzt werden kann
- Für den Freiwilligen soll es möglich sein, den tatsächlich bezahlten Gesamtpreis zu hinterlegen
- Es muss gespeichert werden, welche Einkaufsliste von welchen Hilfesuchenden erstellt und von welchem/r Freiwilligen übernommen wurde
- Sobald eine Einkaufsliste von einem/r Freiwilligen übernommen wurde, wird diese Liste für andere Freiwillige ausgeblendet
- Die Einkaufsliste wird dann vom/von der Freiwilligen als erledigt markiert (alle Artikel besorgt), das UI spiegelt diesen Status für den/die Hilfesuchende wieder
- Benutzeraktionen werden in einem Log erfasst und gespeichert (IP Adresse, Aktion, Benutzername, Timestamp)

1.2 BenutzerInnen / Rollen

Es existieren folgende Benutzerrollen:

1. HilfesuchendeR

Kann Einkaufslisten anlegen (1:n) und verwalten

2. FreiwilligeR

Sieht alle nicht zugeordneten Listen sowie jene, die er/sie schon übernommen hat. Er/sie kann die Einkaufsliste Artikel für Artikel abarbeiten.

HINWEIS: der Registrierungsprozess für BenutzerInnen muss nicht implementiert werden. (BenutzerInnen können direkt in der Datenbank hinterlegt werden.)

2 Non-funktionale Anforderungen

2.1 Programmierung Server

Umsetzung mittels PHP ≥ 7.3 (OOP) – achten Sie auf einen sauberen Aufbau und eine Trennung der einzelnen Anwendungsschichten. Achten Sie auch darauf, die Features von PHP 7 (Typisierung, Operatoren) zu nutzen. Für die Anbindung der Datenquelle benutzen Sie die Bibliothek *PDO*. Sichern Sie Ihre Anwendung gegen SQL-Injection und XSS-Angriffe durch serverseitige Prüfung der Eingaben und Parameter.

Bei Benutzeraktionen werden zusätzlich Daten zur Identifizierung mitgespeichert werden (bspw. IP-Adresse der BenutzerInnen)

2.2 Datenbank

Die Datenverwaltung erfolgt mittels MySQL 5 (InnoDB). Achten Sie auf die Grundregeln des Datenbank-Designs (Datentypen, Normalisierung, sinnvolle Constraints, ...). Hinweis: Datensätze, die von den BenutzerInnen gelöscht wurden, sollten nicht direkt aus der DB gelöscht werden, sondern lediglich mit einem entsprechenden Flag gekennzeichnet werden.

2.3 Client / Frontend

Die Verwendung von Javascript (auch Frameworks, aber lediglich UI Frameworks wie bspw. JQuery UI - keine SPA-Frameworks wie Angular, React oder VueJS) ist erlaubt. Die Logik muss serverseitig implementiert werden. Der HTML-Code sollte HTML5 valide sein, sämtliche Formatierungen via CSS umgesetzt werden. Die Benutzeroberfläche sollte intuitiv bedienbar sein und bei neuen BenutzerInnen der Anwendung keinerlei Erklärungsbedarf erfordern. Achten Sie insbesondere auf Validierung von Eingaben sowie die Fehlerausgabe.

Auch eine gezielte Verwendung von AJAX ist möglich, sofern diese für Sie sinnvoll erscheint (ist jedoch nicht zwingend Voraussetzung).

HINWEIS: das „Design“ sollte einfach, aber funktional sein – legen Sie Augenmerk auf intuitive Bedienbarkeit und logische Führung der BenutzerInnen durch die Website (Navigationselemente, Feedback bei Benutzeraktionen, Formular-Validierungen & -Layout, ...). Achten Sie insbesondere auf eine übersichtliche Darstellung, wenn die Listen länger werden.

3 Form der Abgabe

3.1 Dokumentation

Erstellen Sie für Ihre Datenbank-Umsetzung ein ER- oder UML-Diagramm, welches die Entitäten und Beziehungen visualisiert. Für die Anwendung erstellen Sie bitte eine textuelle Beschreibung und / oder eine dokumentierte Skizze der Architektur.

3.2 Testfälle

Führen Sie ausführliche Tests Ihrer Anwendung durch und dokumentieren Sie diese. Testen Sie auch fehlerhafte Eingaben! Ihre Testfälle sollten auf jeden Fall folgende Zustände testen:

- Benutzerlogin (inkl. Fehlerüberprüfung und -behandlung) der unterschiedlichen Rollen
- Darstellung der Applikation aus Sicht der unterschiedlichen Rollen inkl. Status (offene Listen, eigene Listen, abgearbeitete Listen, bereits erledigte Artikel)
- Hilfesuchender verwaltet Listen (erstellen, ändern, löschen)
- Hinzufügen von Artikeln
- Freiwilliger übernimmt Liste und arbeitet diese ab

Erstellen Sie eine entsprechende Dokumentation und dokumentieren Sie darüber hinaus die Testfälle in Form eines PDF-Dokuments.

3.3 Anwendung

1. Datenbank

SQL – Dump inkl. Testdaten und CREATE DATABASE statement

Datenbankname: **fh_2020_scm4_[IHRE_MATRIKELNUMMER]**
Benutzername: **fh_2020_scm4**
Passwort: **fh_2020_scm4**

2. Code

Zip-Archiv, Startdokument = index.php, muss auf Standard – XAMPP Installation lauffähig sein. Die Dokumentation speichern Sie bitte im Unterverzeichnis /doc.

3.4 Deadline

Laden Sie Ihre Anwendung als **eine Datei im ZIP-Format** in der entsprechenden Abgabe via Moodle hoch. Den konkreten Termin entnehmen Sie bitte dem Abgabe-Modul in Moodle.

3.5 Beurteilung

Die Beurteilung der Übung erfolgt im Rahmen einer Präsentation vor der eigenen Gruppe. Bei dieser Präsentation stellen Sie bitte Ihre Lösung vor, demonstrieren die Testfälle und stehen für technische und inhaltliche Fragen bzw. Code-Reviews zur Verfügung.

Der genaue Termin wird noch kommuniziert.

SCR4 Simple Shopping Projekt

Gerald Rieß

30. Juni 2020

Inhaltsverzeichnis

1	Simple Shopping Dokumentation	5
2	Beschreibung	5
2.1	Allgemein	5
2.2	Hilfesuchender	5
2.3	Freiwilliger	5
2.4	Datenbank	6
2.5	Struktur	6
3	UML-Diagramm	6
4	Testfälle	8
4.1	Hilfesuchender	9
4.2	Freiwilliger	12
4.3	Weitere Fehlerbehandlungen	14

1 Simple Shopping Dokumentation

2 Beschreibung

2.1 Allgemein

Umgesetzt wurde eine Webseite mit der Hilfesuchende Einkaufslisten erstellen können. Ein Freiwilliger kann die Listen übernehmen und die Artikel abarbeiten. Die Benutzer müssen sich authentifizieren um die Anwendung nutzen zu können. Ein Benutzer kann Listen erstellen und diese verwalten. Ein Freiwilliger kann eine Liste abarbeiten und die einzelnen Artikel, die bereits gekauft sind, in der Anwendung abhacken. Wenn der Freiwillige mit der Liste fertig ist, muss dieser den bezahlten Preis eintragen und auf den Button „Erledigt“ klicken. Danach ist die Liste beim Hilfesuchenden und beim Freiwilligen als erledigt gekennzeichnet.

Alle Interaktionen laufen über einen Controller, dies ist der zentrale Knotenpunkt, der die Aktionen abarbeitet. Die Aktionen können so einfach mitgelogged werden und eine Erweiterung der Aktionen ist sehr einfach möglich.

2.2 Hilfesuchender

Nach dem Login befindet sich der Hilfesuchende auf der Startseite, diese ist für jeden Benutzer gleich. Folgende Seiten stehen dem Hilfesuchenden zur Auswahl:

- Startseite
- neue Liste erstellen
- offene Listen ansehen und bearbeiten
- Listen in Arbeit
- geschlossene- bzw. erledigte Listen ansehen

2.3 Freiwilliger

Nach dem Login befindet sich der Hilfesuchende auf der Startseite, diese ist für jeden Benutzer gleich. Folgende Seiten stehen dem Hilfesuchenden zur Auswahl:

- Startseite
- freigegebene Listen von Hilfesuchenden ansehen
- Listen die ausgewählt wurden abarbeiten

- abgearbeitete Listen ansehen

Wird der User in einer anderen Session abgemeldet, muss er sich neu einloggen. Wenn ein anderer User angemeldet wird und eine Liste hinzufügen möchte, dieser aber nicht berechtigt ist, wird dies mit einer Fehlermeldung angezeigt.

2.4 Datenbank

Die Anwendung beinhaltet eine Datenbank, die die Daten verwaltet. Die beiden geforderten Benutzer befinden sich bereits im Datenbank DUMP-File und sind nach dem Einspielen des DUMP bereits vorhanden. Jeder User verfügt über eine Rolle die Auskunft darüber gibt, ob der Benutzer ein Hilfesuchender oder ein Freiwilliger ist. Dank der Rollen ist es möglich verschiedene Views anzuzeigen, die auf die Benutzer zugeschnitten sind. Einige andere Mock-Daten wie Listen und Artikel sind schon vorhanden. Die Datenbank beinhaltet alle geforderten Entitäten. Daten können von der Datenbank nicht wirklich gelöscht werden. Ich habe wie gefordert Flags verwendet damit ich eruieren kann, ob eine Liste oder ein Artikel noch angezeigt werden soll oder nicht. Es wurden stets prepared statements verwendet um SQL-Injection zu verhindern. Außerdem werden Aus- und Eingaben der Datenbank escaped, damit cross side scripting unterbunden wird.

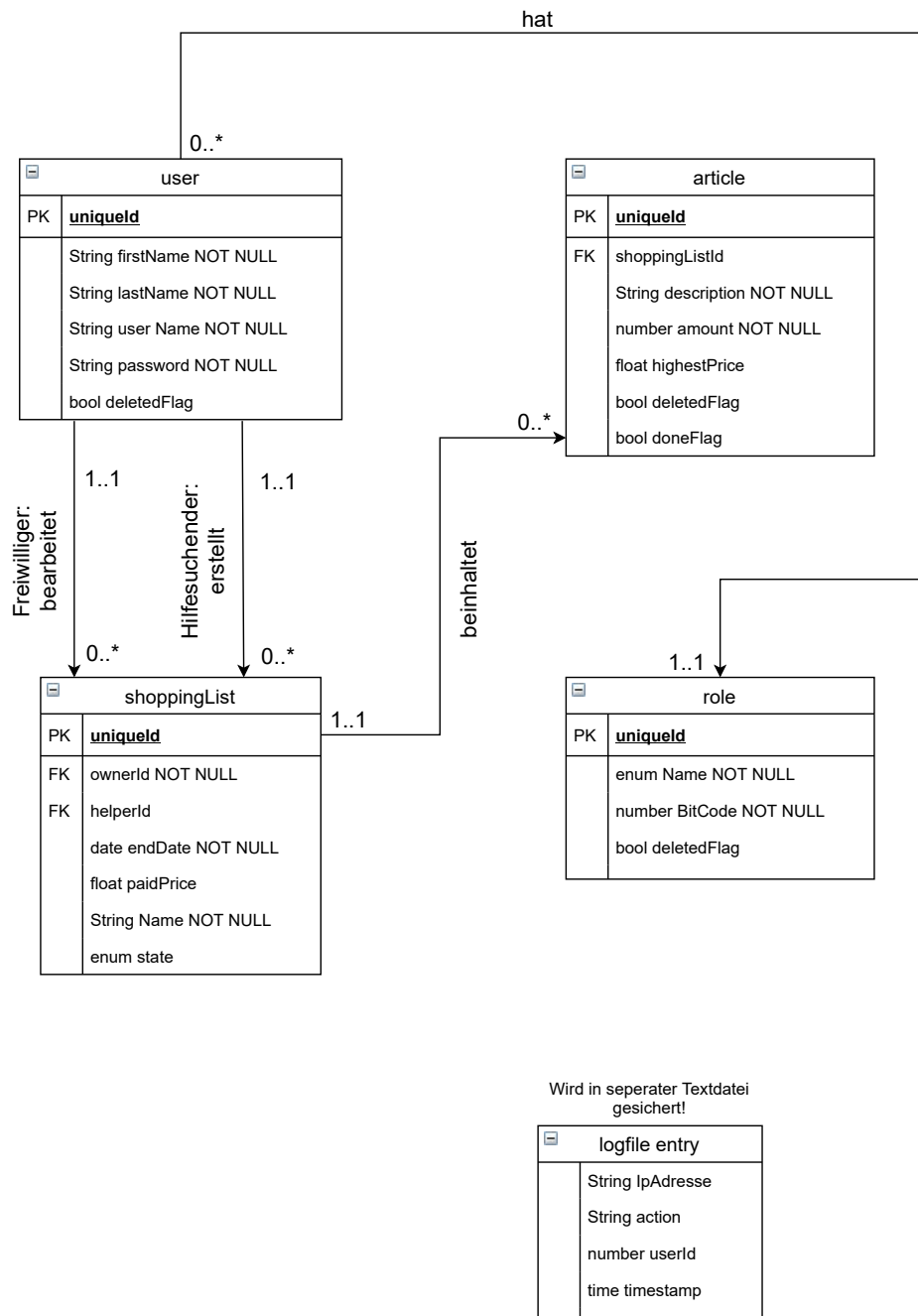
2.5 Struktur

Verwendet wird ein Model View Controller Pattern. Alle clientspezifischen Seiten oder Skripts sind unter Views vorhanden. Der Code für den Server befindet sich im Ordner Lib. Generell werden nur Requests zwischen den beiden Schichten ausgetauscht. Der Controller verwaltet alle Actions und bildet die Schnittstelle zwischen Client und Server.

3 UML-Diagramm

Das UML Diagramm, welches die Entitäten und Beziehungen festhält, wurde mit Draw-IO erstellt. Auf Basis dieses Diagrammes habe ich meine Datenbank erstellt.

WEBSHOP UML -DIAGRAMM



4 Testfälle

Hier die Loginseite die ein User erhält der die Homepage öffnet.

Simple Shopping Startseite Nicht eingeloggt

Authentifizierung

Benutzername:

Passwort:

Login

Abbildung 1: Loginseite

Bei fehlerhaftem Login wird folgende Seite ausgegeben:

Authentifizierung

Benutzername:

Passwort:

Login

• Falscher Beutzername oder Passwort

Abbildung 2: Fehleingabe User

4.1 Hilfesuchender

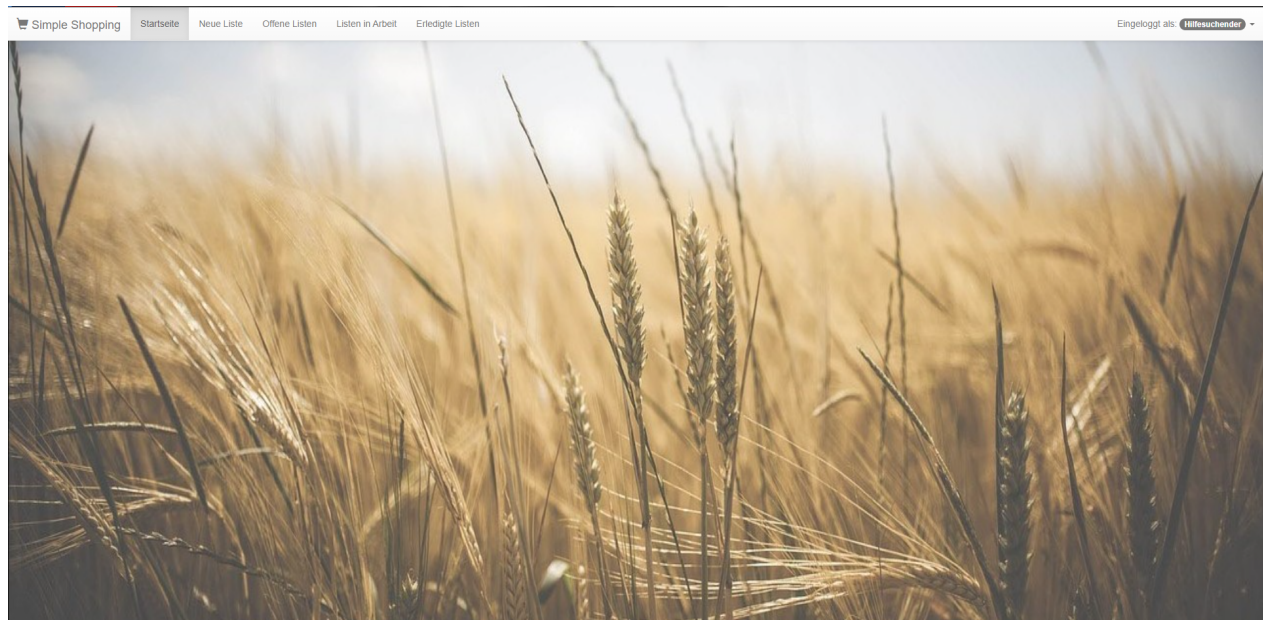


Abbildung 3: Startseite Hilfesuchender

Hier wird die Einkaufsliste vom Hilfesuchenden erstellt. Dieser muss nur einen Namen und ein Enddatum eingeben, bis wann die Liste abgearbeitet sein soll.

The image shows a web form titled 'Einkaufsliste erstellen'. Below the title is a light gray box with the text 'Please fill out the form below:'. Inside this box, there are two input fields. The first is labeled 'Name:' and contains the text 'Grillen'. The second is labeled 'Enddatum:' and contains the date '06.2020'. To the right of the date is a small calendar icon. Below these fields is a button labeled 'Erstellen'.

Abbildung 4: Einkaufsliste erstellen

Fügt man eine Liste hinzu wird man sofort an die Seite der offenen Listen weitergeleitet. Hier kann man mit dem Klick auf den oberen grünen Button die Liste bearbeiten. Mit dem Klick auf den unteren grünen Button wird die Liste gelöscht. Beim Klick auf den Button Freigeben wird die Liste für alle Hilfesuchenden angezeigt.

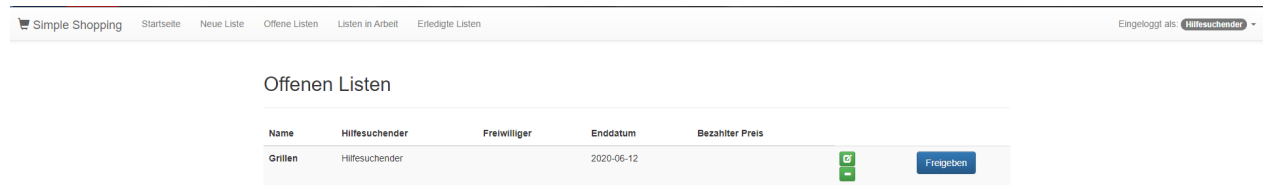


Abbildung 5: Einkaufsliste erstellt

Dies ist die Bearbeitungsansicht der Liste. Hier werden Artikel hinzugefügt. Bei Artikeln müssen der Name, die Menge und der Höchstpreis angegeben werden. Bei der Menge und Preisangabe ist es natürlich nicht möglich Buchstaben einzugeben.

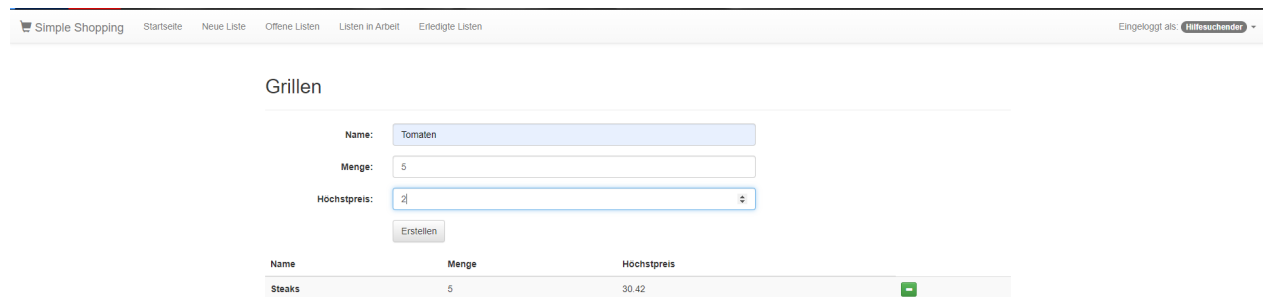


Abbildung 6: Einkaufsliste bearbeiten

Wird doch ein Buchstabe eingegeben erhält man eine entsprechende Meldung.

Grillen

Name:

Menge:

Höchstpreis:

! Geben Sie eine Nummer ein.

Name	Menge	Höchstpreis
Steaks	5	30.42

Abbildung 7: Einkaufsliste bearbeiten Fehleingabe

In der folgenden Abbildung ist das Steak gelöscht worden und es befinden sich nur noch die

Tomaten, die zuvor richtig eingetragen wurden, in der Liste.

Simple Shopping Startseite Neue Liste Offene Listen Listen in Arbeit Erledigte Listen Eingeloggt als **Hilfesuchender**

Grillen

Name:

Menge:

Höchstpreis:

Name	Menge	Höchstpreis
Tomaten	2	2

Abbildung 8: Einkaufsliste Artikel Steak gelöscht

Wenn man mit dem Bearbeiten fertig ist, kann mit einem Klick auf den Button Öffene Listen in der Navbar wieder zur Ansicht der offenen Listen gelangen. Die Liste kann nun mit dem Button Freigeben veröffentlicht werden.

Simple Shopping Startseite Neue Liste Offene Listen Listen in Arbeit Erledigte Listen Eingeloggt als **Hilfesuchender**

Offenen Listen

Name	Hilfesuchender	Freiwilliger	Enddatum	Bezahlt Preis
Grillen	Hilfesuchender		2020-06-12	<input type="button" value="Freigeben"/>

Abbildung 9: Einkaufsliste freigegeben

Ist die Liste veröffentlicht wird der Button grau und der Text freigegeben erscheint.

Simple Shopping Startseite Neue Liste Offene Listen Listen in Arbeit Erledigte Listen Eingeloggt als **Hilfesuchender**

Offenen Listen

Name	Hilfesuchender	Freiwilliger	Enddatum	Bezahlt Preis
Grillen	Hilfesuchender		2020-06-12	<input type="button" value="Freigegeben"/>

Abbildung 10: Einkaufsliste freigegeben

Die Testliste wird hier mit dem Klick auf den grünen Minus Button gelöscht.

Simple Shopping

Startseite

Neue Liste

Offene Listen

Listen in Arbeit

Erledigte Listen

Eingeloggt als:

Hilfesuchender

Offenen Listen

Name	Hilfesuchender	Freiwilliger	Enddatum	Bezahlter Preis		
Testliste	Hilfesuchender		2020-06-10		<div><div></div><div></div></div>	<div>Freigeben</div>
Grillen	Hilfesuchender		2020-06-12		<div><div></div><div></div></div>	<div>Freigeben</div>

Abbildung 11: Einkaufsliste löschen

Ergebnis des Löschvorgangs. Testliste nicht mehr vorhanden.

Simple Shopping

Startseite

Neue Liste

Offene Listen

Listen in Arbeit

Erledigte Listen

Eingeloggt als:

Hilfesuchender

Offenen Listen

Name	Hilfesuchender	Freiwilliger	Enddatum	Bezahlter Preis		
Grillen	Hilfesuchender		2020-06-12		<div><div></div><div></div></div>	<div>Freigeben</div>

Abbildung 12: Einkaufsliste gelöscht

4.2 Freiwilliger

Ansicht der veröffentlichten Listen

Simple Shopping

Startseite

Freigegebene Listen

Abzuarbeitende Listen

Abgearbeitete Listen

Eingeloggt als:

Freiwilliger

Alle freigegebenen Listen

Name	Hilfesuchender	Freiwilliger	Enddatum	Bezahlter Preis	
Grillen	Hilfesuchender		2020-06-12		<div><div></div><div>Übernehmen</div></div>

Abbildung 13: Einkaufslisten freigegeben

Listen können, bevor Sie übernommen werden, betrachtet werden. Danach können Sie mit dem "Übernehmen Button" übernommen werden.

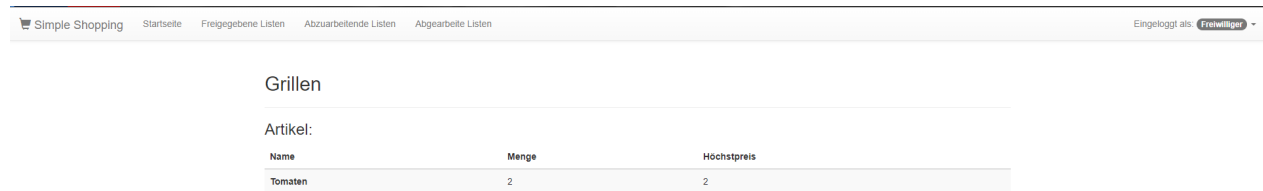


Abbildung 14: Einkaufsliste betrachten

Die übernommenen Listen sind in der Ansicht abzuarbeitende Listen. Mit dem Druck auf den grünen Bearbeitungsbutton kommt man in eine Bearbeitungsansicht. Hier können die Artikel betrachtet werden.

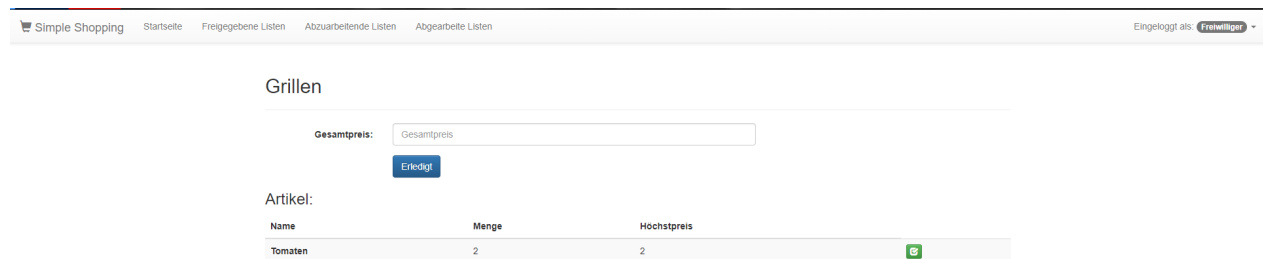


Abbildung 15: Einkaufsliste abarbeiten

Ist ein Artikel besorgt, kann der Freiwillige diesen abhaken. Dies wird durch einen blau eingefärbten check Button angezeigt. Ist man mit dem Einkauf fertig, wird oben der Gesamtpreis eingetragen. Mit dem Klick auf den Button erledigt ist die Liste fertig abgearbeitet.

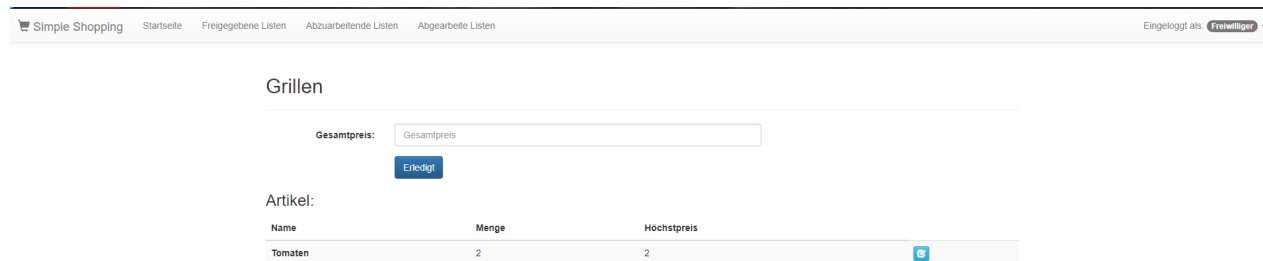
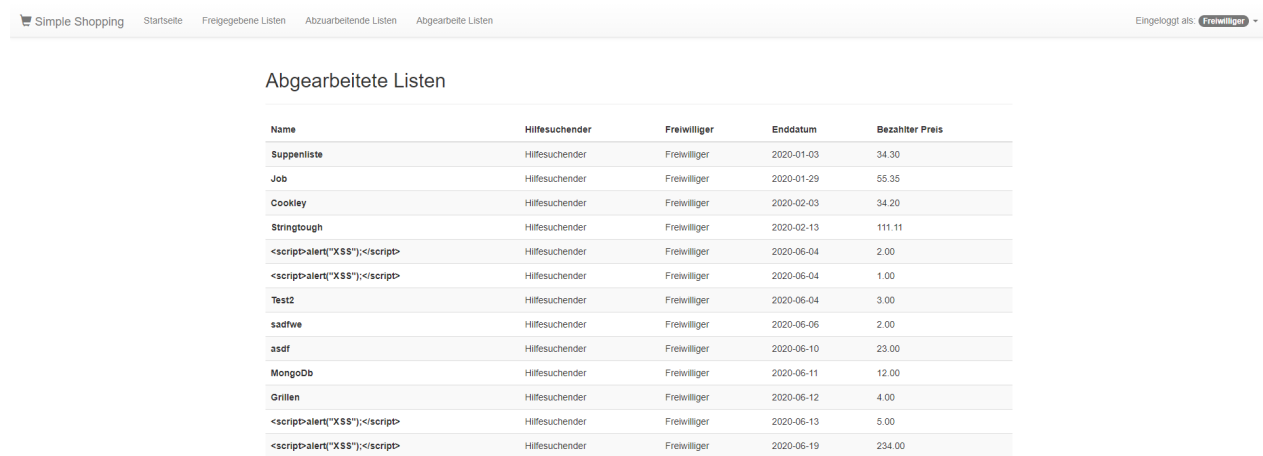


Abbildung 16: Einkaufsliste abarbeiten Artikel abhaken

Hier noch die Ansicht der abgearbeiteten Listen des Freiwilligen.

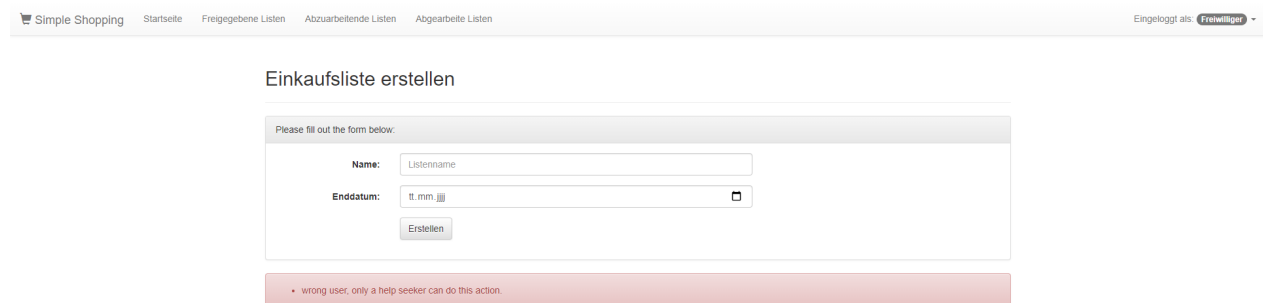


Name	Hilfesuchender	Freiwilliger	Enddatum	Bezahlter Preis
Suppenliste	Hilfesuchender	Freiwilliger	2020-01-03	34.30
Job	Hilfesuchender	Freiwilliger	2020-01-29	55.35
Cookie	Hilfesuchender	Freiwilliger	2020-02-03	34.20
Stringtough	Hilfesuchender	Freiwilliger	2020-02-13	111.11
<script>alert('XSS');</script>	Hilfesuchender	Freiwilliger	2020-06-04	2.00
<script>alert('XSS');</script>	Hilfesuchender	Freiwilliger	2020-06-04	1.00
Test2	Hilfesuchender	Freiwilliger	2020-06-04	3.00
sadfwe	Hilfesuchender	Freiwilliger	2020-06-06	2.00
asdf	Hilfesuchender	Freiwilliger	2020-06-10	23.00
MongoDb	Hilfesuchender	Freiwilliger	2020-06-11	12.00
Grillen	Hilfesuchender	Freiwilliger	2020-06-12	4.00
<script>alert('XSS');</script>	Hilfesuchender	Freiwilliger	2020-06-13	5.00
<script>alert('XSS');</script>	Hilfesuchender	Freiwilliger	2020-06-19	234.00

Abbildung 17: Einkaufsliste abgearbeitete Listen

4.3 Weitere Fehlerbehandlungen

In diesem wurde in einem zweiten Tab der Hilfesuchende ausgeloggt. Ein Freiwilliger hat sich anstelle dessen eingeloggt und im alten Tab wird auf Liste hinzufügen geklickt. Dies führt dazu, dass eine Fehlermeldung erscheint, weil der falsche Benutzer angemeldet ist. Die Seite bleibt zwar geöffnet, der Freiwillige kann aber nichts manipulieren. Zu einer alten URL eines Hilfesuchenden zurückzugelangen ist generell möglich. Aktionen können aber nicht ausgeführt werden vom Freiwilligen.



Please fill out the form below:

Name:

Enddatum:

• wrong user, only a help seeker can do this action.

Abbildung 18: Falscher Nutzer möchte Liste anlegen