

SHAREDSHOPPING

SCM4 / SS 2020

Christoph Haider

FH Hagenberg SE BB | S1810307013

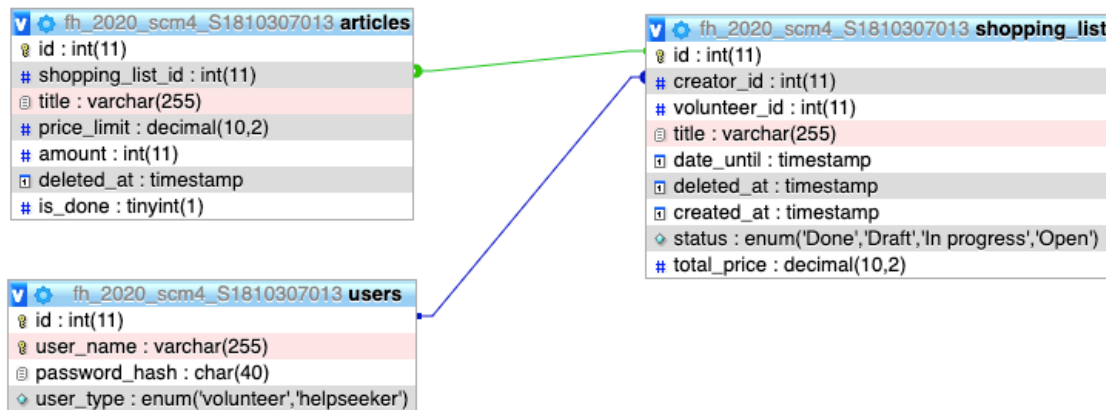
Inhalt

1	Datenbank Diagramm.....	2
2	Architektur.....	2
2.1	Namespace SharedShopping	2
2.2	Namespace Data	3
2.3	Namespace Logging.....	3
2.4	Frontend	4
3	Anwendungsablauf.....	4
4	Screenshots.....	4
5	Tests	9
5.1	Benutzerlogin und Dashboard aus Sicht unterschiedlicher Rollen	10
5.1.1	Falsche Daten	10
5.1.2	Login als Volunteer	10
5.1.3	Login als Helpseeker	10
5.2	Shopping List Details	11
5.2.1	Helpseeker Sicht (Draft)	11
5.2.2	Helpseeker Sicht (Published)	12
5.2.3	Volunteer Sicht (Open)	12
5.2.4	Volunteer Sicht (In progress)	13
5.2.5	Volunteer Sicht (Alle Artikel abgehakt)	13
5.2.6	Volunteer Sicht (Done)	14
5.2.7	Helpseeker Sicht (Done)	14
5.3	Workflow Tests	14
5.3.1	Frontend Validierung.....	14
5.3.2	Backend Validierung.....	15
5.3.3	Shopping List Status	15
5.4	Errors	16

1 Datenbank Diagramm

Das SQL-File „etc/shared-shopping.sql“ baut die Datenbank, einen Datenbank Nutzer und alle benötigten Tabellen auf. Außerdem befinden sich INSERT Statements im File um die Datenbank mit Testdaten und Benutzern zu füllen.

Prinzipiell werden nur 3 Tabellen benötigt: **shopping_list**, **articles** und **users**.



2 Architektur

Die File index.php dient als Router für alle Views der Anwendung. Im Ordner „views“ befinden sich die Hauptviews, diese nutzen weitere Views aus dem „views/partials“ Ordner.

Im „partials“ Ordner befinden sich header und footer templates. Außerdem gibt es für beide Nutzerrollen jeweils eine dashboardView und eine detailView mit verschiedenen Implementierungen.

Um Codestücke wiederverwenden zu können wurden Inputs und Input Gruppierungen für Artikel und Shopping Lists ebenfalls in Templates gepackt.

Der „lib“ Ordner hält die gesamte Logik der Anwendung.

2.1 Namespace SharedShopping

Grundsätzlich

Klassen für Article, ShoppingList und User um deren Objekte halten zu können. Zusätzlich Enums für UserType und ShoppingListStatus.

Controller

Herzstück der Anwendung. Verwaltet alle POST Requests die durch Forms gefeuert werden. Bevor eine Operation beginnt wird die Authentifizierung des aktuellen Users geprüft. Zusätzlich wird mit Hilfe des PermissionManagers geprüft ob die Aktion durch den User

umgesetzt werden darf. Übergebene Datensätze werden durch einen ValidationManager validiert (zusätzlich zur Frontend Validation notwendig).

Wenn eine der obigen Bedingungen (Authentifiziert, Permission, Validation) nicht gegeben ist, wird erst in das Log File geschrieben, anschließend wird der jeweilige Error angezeigt.

Sind die Bedingungen erfüllt und die Operation war erfolgreich wird ebenfalls geloggt und anschließend wird der User weitergeleitet (entweder auf dieselbe Seite oder eine andere Destination). Die erwähnten Operationen sind üblicherweise CRUD Operationen auf die Datenbank.

PermissionManager

Bietet mehrere Methoden, die anhand einer übergebenen Shopping List ID prüfen, ob bestimmte Aktionen durch den eingeloggten User erlaubt sind. Dazu gehört auch die Prüfung, ob beispielsweise eine Shopping List ihren Status in einen anderen Zielstatus ändern kann.

ValidationManager

Prüft durch Regular Expressions, ob übergebene Datensätze der richtigen Form entsprechen und nur erlaubte Zeichen enthalten.

AuthenticationManager

Authentifiziert beim Login und bietet Methoden zur Prüfung der Authentifizierung. Bietet auch eine Methode zum Erhalt des UserType des eingeloggten Users.

Die Session hält dabei lediglich ein Property ‚user‘ mit der User ID als Wert.

2.2 Namespace Data

DataManager

Ist als Singleton implementiert und baut die Verbindung zur Datenbank auf.

Statements werden als prepared Statements abgesetzt. Bietet somit Schutz gegen SQL Injection (neben der Validierung der Daten im Controller).

Bietet CRUD Operationen auf die Datenbank für User, Shopping Lists und Articles. Richtige DELETE Statements existieren nicht, Tabellen Reihen werden lediglich durch ein **delete_at** Flag (enthält DateTime) als gelöscht markiert.

Eine gelöschte Shopping List führt zur Löschung ihrer Artikel.

2.3 Namespace Logging

FileLogger

Klasse zum Loggen von Daten in ein Log-File. Die Implementierung wurde dabei größtenteils von einem öffentlichen Github Repo übernommen (Das Rad muss nicht neu erfunden werden).

Bietet die Möglichkeit zwischen Arten von Loggings zu unterscheiden (NOTICE; WARNING, ERROR, FATAL).

Die Logdaten werden in /logs/log.log gespeichert.

2.4 Frontend

Für das Frontend wurde Bootstrap CSS verwendet. Generell wurde darauf geachtet, dass Buttons gut deutbar sind anhand ihrer Symbole und Farben. Das Layout ist schlicht gehalten und sollte übersichtlich sein. Wiederkehrende Elemente wurden in Template Dateien verpackt. Zur Kommunikation von Frontend und Backend wurden vor allem Forms verwendet (Forderung der Übung gesamte Logik in PHP zu halten).

3 Anwendungsablauf

Hilfesuchende/r erstellt eine neue Liste im Dashboard und vergibt dabei Listen Name und Enddatum. Als nächstes kann der User mehrere Artikel erstellen, jeweils mit Bezeichnung, Anzahl und Preis Limit. Diese Artikel, sowie auch die Listen Daten können verändert bzw. gelöscht werden.

Ist die Liste fertig kann sie freigegeben werden, und ist somit für Freiwillige sichtbar.

Das Dashboard gibt Überblick für beide Benutzerrollen, welche Listen in welchen Status haben.

Übernehmen Freiwillige nun eine offene Liste wird diese auf IN_PROGRESS gesetzt und ist für andere Freiwillige nicht mehr sichtbar. Die Liste kann nun abgearbeitet werden. Ist der letzte Artikel abgeharkt muss noch der tatsächlich bezahlte Gesamt Preis angegeben werden um die Liste anschließend als „Done“ markieren zu können.

Beide Nutzerrollen wissen zu jedem Zeitpunkt über den Status der Liste Bescheid. Hilfesuchende sehen außerdem wer die Einkaufsliste übernommen hat.

4 Screenshots

Shared Shopping

Home

Dashboard

Logged in as **Hans_Peter** ▾

Welcome to Shared Shopping

Please log in to use the full functionality of Shared Shopping

I look for help with my groceries!

With this software you can create as many shopping lists as you need. Just set the date until you expect the articles to be delivered to you. You can add as many articles to the list as you wish. Just provide the name, the amount and maximum price limit you are willing to pay. One of the many Volunteers will take over your list and get all the articles. When the volunteer is done you'll see the status of your shopping list in the Details View.

I want to help others!

Just log in and choose one of the OPEN shopping lists. After you picked one, it will show up in the IN PROGRESS area of your dashboard. By modifying it you can easily see all the information you need. Just check all the articles you already bought. When you got all the products you have to enter the total price you paid to finish the shopping list.

Shared Shopping

Home

Dashboard

Logged in as **Hans_Peter** ▾

Dashboard

Create New Shopping List

26.06.2020

Create

Your Shopping Lists

Drafts

Title	Date Until	Status	Action
My second list	2020-06-16	Draft	Modify ⚙
This is my first list	2020-06-27	Draft	Modify ⚙
asdfasdfasdf	2020-06-26	Draft	Modify ⚙

Open

Title	Date Until	Status	Action
My first list	2020-06-16	Open	Modify ⚙

In Progress

Title	Date Until	Status	Action
My third list	2020-06-16	In progress	Modify ⚙

Done

No shopping lists found for this status

Shared Shopping

Home

Dashboard

Logged in as Hans_Peter

This is my first list

Draft

Delete Shopping List

Publish Shopping List

Shopping List Data

This is my first list

27.06.2020

Save Changes

Articles

Article Name	Amount	Price Limit	Action
Milch	1	1.25	<div><div></div><div></div></div>
Brot	1	2.5	<div><div></div><div></div></div>
Nutella	2	10.2	<div><div></div><div></div></div>
Name	Amount	Price	<div><div></div></div>

Shared Shopping

Home

Dashboard

Logged in as Hans_Peter

This is my first list

Open

Delete Shopping List

Shopping List Data

This is my first list

26.06.2020

Save Changes

Articles

Article Name	Amount	Price Limit	Action
Milch	1	1.25	
Brot	1	2.5	
Nutella	2	10.2	

Shared Shopping

Home

Dashboard

Logged in as VolunteerByHeart

My first list

Open

Take this list

Shopping List Data

My first list

16.06.2020

Articles

Article Name	Amount	Price Limit	Action
Brot 1kg	1	2.95	
Cornetto Heidelberg	4	5.65	

Shared Shopping

Home

Dashboard

Logged in as VolunteerByHeart

My first list

In progress

Shopping List Data

My first list

26.06.2020

Articles

Article Name	Amount	Price Limit	Action
Brot 1kg	1	2.95	✓
Cornetto Heidelberg	4	5.65	✓

Shared Shopping

Home

Dashboard

Logged in as VolunteerByHeart

My first list

In progress

24.25

Finish this list

Shopping List Data

My first list

26.06.2020

Articles

Article Name	Amount	Price Limit	Action
Brot 1kg	1	2.95	<div>✕</div>
Cornetto Heidelberg	4	5.65	<div>✕</div>

Shared Shopping

Home

Dashboard

Logged in as VolunteerByHeart

My first list

Done

24.25

Finish this list

Shopping List Data

My first list

26.06.2020

Articles

Article Name	Amount	Price Limit	Action
Brot 1kg	1	2.95	
Cornetto Heidelberg	4	5.65	

Shared Shopping

Home

Dashboard

Logged in as **Hans_Peter**

My first list

Volunteer: VolunteerByHeart

Total Price: 24.25 €

Done

Delete Shopping List

Shopping List Data

My first list

26.06.2020

Save Changes

Articles

Article Name	Amount	Price Limit	Action
Brot 1kg	1	2.95	
Cornetto Heidelbeer	4	5.65	

5 Tests

Da der Registrierungsvorgang nicht implementiert werden muss, werden insgesamt 4 User bereitgestellt.

User Name	Passwort	User Type
Hans Peter	test	HELPSEEKER
Traudi	test	HELPSEEKER
SuperGuy95	test	VOLUNTEER
VolunteerByHeart	test	VOLUNTEER

5.1 Benutzerlogin und Dashboard aus Sicht unterschiedlicher Rollen

5.1.1 Falsche Daten

Shared Shopping Home Dashboard Not logged in! ▾

• Invalid user name or password.

Login

Please fill out the form below:

User name:

Password:

5.1.2 Login als Volunteer

Nur offene und angenommene Listen sind sichtbar.

Shared Shopping Home Dashboard Logged in as SuperGuy95 ▾

Dashboard

Your Volunteer Shopping Lists

Open

Title	Date Until	Status	Action
This is my first list	2020-06-26	Open	Modify ⚙️

In Progress

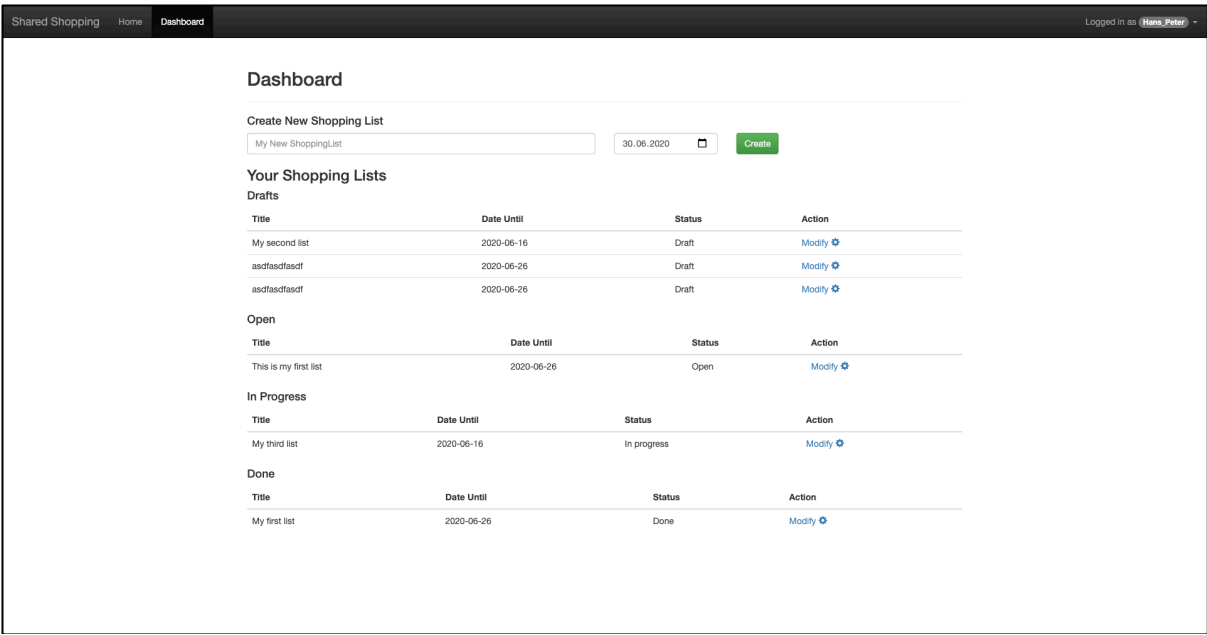
Title	Date Until	Status	Action
My third list	2020-06-16	In progress	Modify ⚙️

Done

No shopping lists found for this status

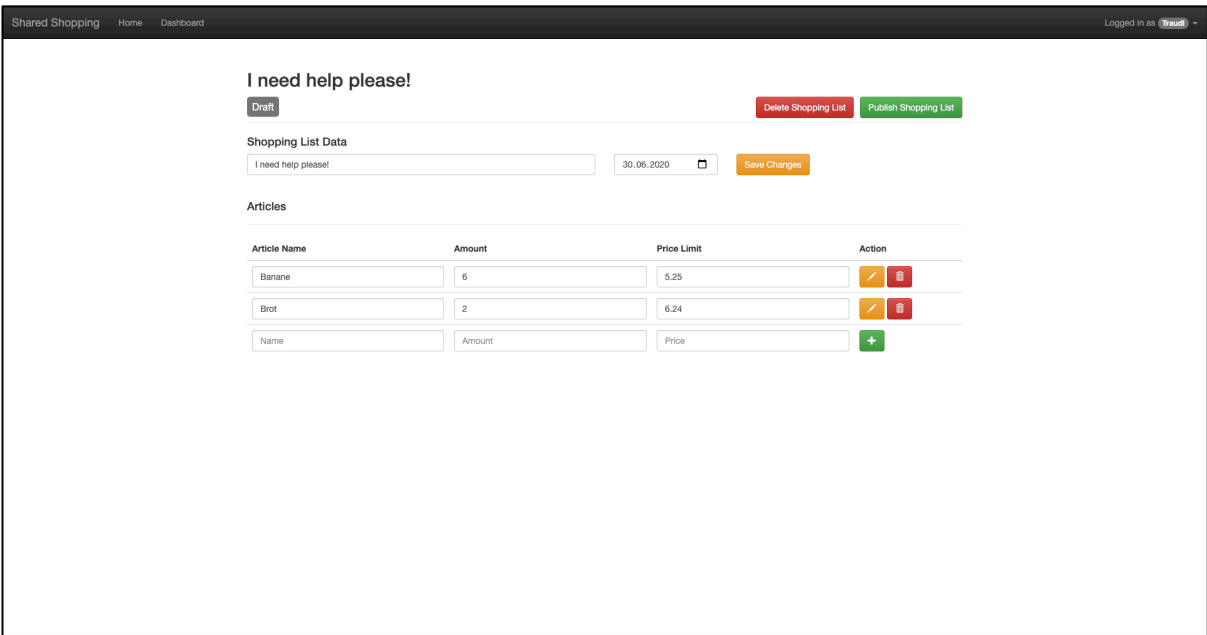
5.1.3 Login als Helpseeker

Nur die eigenen Listen sind sichtbar.



5.2 Shopping List Details

5.2.1 Helpseeker Sicht (Draft)



5.2.2 Helpseeker Sicht (Published)

Shared Shopping

Home

Dashboard

Logged in as **Traudl**

I need help please!

Open

Delete Shopping List

Shopping List Data

I need help please!

30.06.2020

Save Changes

Articles

Article Name	Amount	Price Limit	Action
Banane	6	5.25	
Brot	2	6.24	

5.2.3 Volunteer Sicht (Open)

Shared Shopping

Home

Dashboard

Logged in as **SuperGuy99**

I need help please!

Open

Take this list

Shopping List Data

I need help please!

30.06.2020

Articles

Article Name	Amount	Price Limit	Action
Banane	6	5.25	
Brot	2	6.24	

5.2.4 Volunteer Sicht (In progress)

Shared Shopping

Home

Dashboard

Logged in as **SuperGuy99**

I need help please!

In progress

Shopping List Data

I need help please!

30.06.2020

Articles

Article Name	Amount	Price Limit	Action
Banane	6	5.25	✓
Brot	2	6.24	✓

5.2.5 Volunteer Sicht (Alle Artikel abgeharkt)

Shared Shopping

Home

Dashboard

Logged in as **SuperGuy99**

I need help please!

In progress

Total Price

Finish this list

Shopping List Data

I need help please!

30.06.2020

Articles

Article Name	Amount	Price Limit	Action
Banane	6	5.25	✗
Brot	2	6.24	✗

5.2.6 Volunteer Sicht (Done)

Shared Shopping Home Dashboard Logged in as **SuperGuy95**

I need help please!

Done 12.21 **Finish this list**

Shopping List Data

I need help please! 30.06.2020

Articles

Article Name	Amount	Price Limit	Action
Banane	6	5.25	
Brot	2	6.24	

5.2.7 Helpseeker Sicht (Done)

Shared Shopping Home Dashboard Logged in as **troub**

I need help please!

Volunteer: SuperGuy95
Total Price: 12.21 €

Done **Delete Shopping List**

Shopping List Data

I need help please! 30.06.2020 **Save Changes**

Articles

Article Name	Amount	Price Limit	Action
Banane	6	5.25	
Brot	2	6.24	

5.3 Workflow Tests

Da sich Workflows schwierig mit Screenshots abbilden lassen werden sie textuell beschrieben.

5.3.1 Frontend Validierung

Bei Erstellung/Änderung der Shopping List Daten, sowie bei Erstellen/Ändern von Artikeln greift die Frontend Validierung. Felder dürfen nicht leer bleiben, Felder die Zahlen oder Preise enthalten werden auch richtig validiert. Dazu werden Browser spezifische, native Warnungen angezeigt.

Shared Shopping Home Dashboard Logged in as **Hans_Peter**

asdfasdfasdf

Draft Delete Shopping List Publish Shopping List

Shopping List Data

asdfasdfasdf 26.06.2020 Save Changes

Articles

Article Name	Amount	Price Limit	Action
asdfasdfasdf	asdf	3	Edit Delete
qwqwew	Please match the format requested.	2	Edit Delete
Name	Amount	Price	+

5.3.2 Backend Validierung

Alle Aktionen die durch Forms ausgelöst werden, sind in Authentifizierung, Permission und (wenn nötig) in Validierungs Checks gewrappt. Bei mehrmaligen wechseln des Users in der selben Session, können Authentifizierung und Permission Checks nicht umgangen werden.

Shared Shopping Home Dashboard Logged in as **Hans_Peter**

No permission to change state of shopping list

This is my first list

Open Delete Shopping List

Shopping List Data

This is my first list 26.06.2020 Save Changes

Articles

Article Name	Amount	Price Limit	Action
Milch	1	1.25	
Brot	1	2.5	
Nutella	2	10.2	

5.3.3 Shopping List Status

Die verschiedenen Status der Shopping Lists können nur in eine Richtung durchlaufen werden, ein Weg zurück existiert nicht.

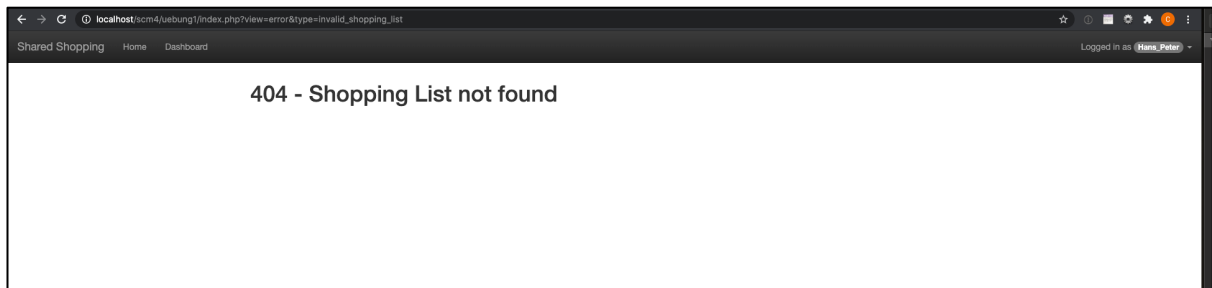
Draft => Open => In Progress => Done

5.4 Errors

Beispielsweise bei Angabe einer Shopping List Id in der URL an den Parameter, die nicht existiert, wird der Fehler behandelt.

Beispiel mit ID 25 (diese existiert nicht in der DB):

<http://localhost/scm4/uebung1/index.php?view=detail&shoppingListId=25>



Nicht eingeloggt auf eine Detail Page per URL steuern

