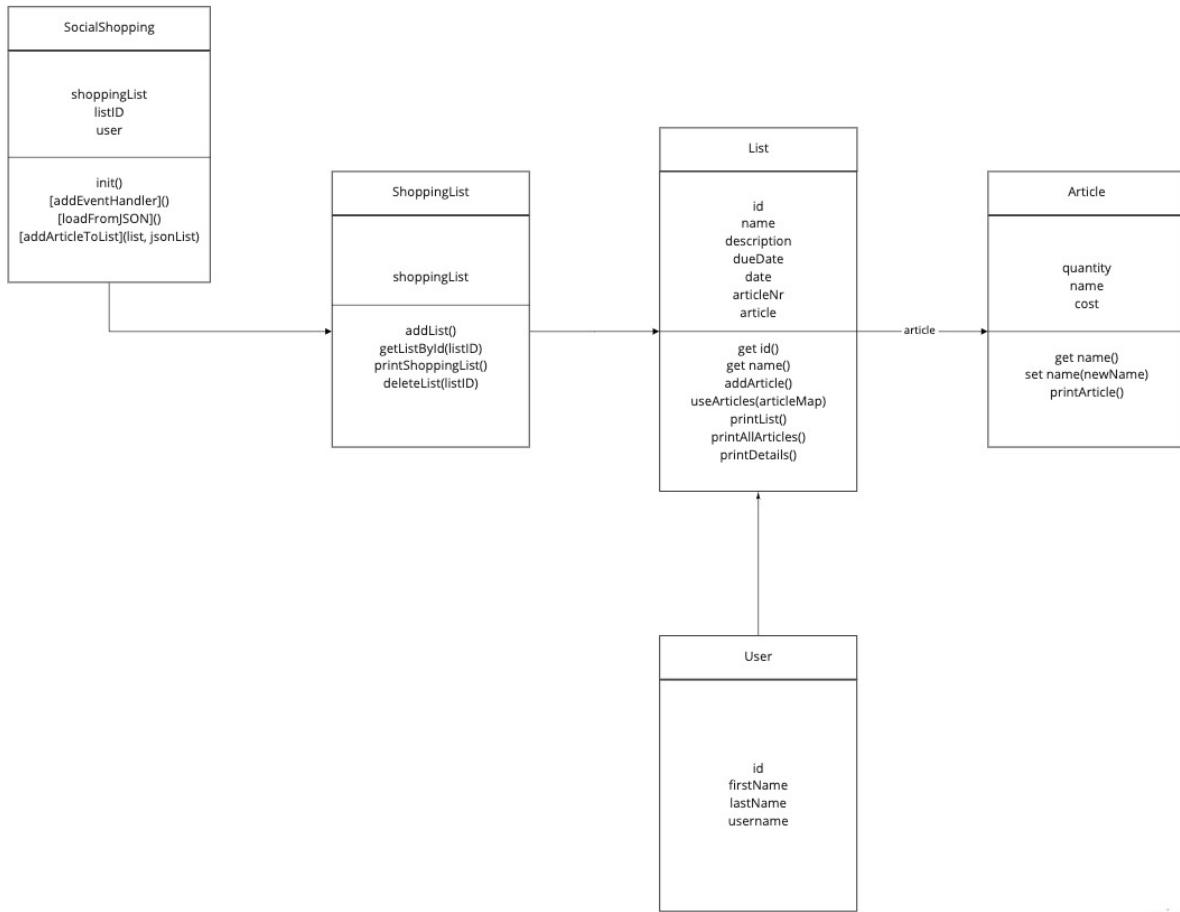


## COVID Social Shopping

### UML-Diagramm



miro

### Sourcecode - index.html:

```

<!doctype html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0,
          maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/
bootstrap.min.css" rel="stylesheet" integrity="sha384-
giJF6kkqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAAsrekwKmP1"
crossorigin="anonymous">
    <link rel="stylesheet" href="style.css">
    <link href="https://fonts.googleapis.com/css2?family=Raleway&display=swap"
rel="stylesheet">
  
```

```
<title>COVID Social Shopping</title>

</head>
<body>
<header>
    <nav>
        <a href="#">Home</a>
        <a href="index.html">Einkaufslisten</a>
        <a href="#">Chat</a>
        <a href="#">Anmelden</a>
        <a href="#">Registrieren</a>
    </nav>

    <div id="btn-help">
        <button id="help" class="topbuttons">Hilfesuchender</button>
        <button id="helper" class="topbuttons">Helfender</button>
    </div>

    <h1>COVID Social Shopping</h1>

    <div id="newList">
        <button id="addList" data-toggle="modal" href="#addListModal"
class="topbuttons">Neue Liste hinzufügen</button>
    </div>
</header>

<div class="list"></div>

<div class="hide" id="detailview">
    <div id="detail">
        <div id="details"></div>
        <table>
            <thead class="tablehead">
                <tr>
                    <th>Anzahl</th>
                    <th>Titel</th>
                    <th>max. Kosten in €</th>
                
```

```
</tr>
</thead>
<tbody class="tablebody"></tbody>
</table>

</div>
<div id="back">
    <button id="goBack" class="topbuttons">Zurück</button>
    <button id="editArticles" class="topbuttons">Bearbeiten</button>
</div>
</div>
```

```
<div class="modal fade" id="addListModal" tabindex="-1" role="dialog" aria-labelledby="addListLabel" aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="addListLabel">Einkaufsliste hinzufügen</h5>
                <button type="button" class="close withoutBorder" data-dismiss="modal" aria-label="Close">
                    <span aria-hidden="true">&times;</span>
                </button>
            </div>
            <div class="modal-body">
                <form>
                    <div class="form-group">
                        <label for="nameCreatorAdd" class="col-form-label">Name des Erstellers:</label>
                        <input type="text" class="form-control" id="nameCreatorAdd">
                    </div>
                    <div class="form-group">
                        <label for="dateCreationAdd" class="col-form-label">Erstellungsdatum:</label>
                        <span id ="dateCreationAdd"></span>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>
```

```
<div class="form-group">
    <label for="titleListAdd" class="col-form-label">Titel
der Einkaufsliste:</label>
    <input type="text" class="form-control"
id="titleListAdd">
</div>

<div class="form-group">
    <label for="descriptionAdd" class="col-form-
label">Beschreibung:</label>
    <input type="text" class="form-control"
id="descriptionAdd">
</div>

<div class="form-group">
    <label for="dueDateAdd" class="col-form-
label">Gewünschtes Erfüllungsdatum:</label>
    <input type="text" class="form-control" id="dueDateAdd">
</div>

<div class="form-group">
    <label for="ArticleAdd" class="col-form-
label">Artikel:</label>
    <input type="text" class="form-control" id="ArticleAdd"
placeholder="Artikelbezeichnung"><br>
    <input type="text" class="form-control"
id="ArticleAddQuantity" placeholder="Menge"><br>
    <input type="text" class="form-control"
id="ArticleAddPrice" placeholder="maximaler Preis">
</div><br>

<a href="#" id="addArticleToList">Artikel hinzufügen</a>
<div id="forArticles">

</div>
</form>
</div>
<div class="modal-footer">
    <a href="#" id="closeChangesAdd" data-
dismiss="modal">Abbrechen</a>
    <a href="#" id="listAdd" data-dismiss="modal">Liste hinzufügen</
a>
```

```
</div>
</div>
</div>
</div>
```

```
<script src="https://code.jquery.com/jquery-3.2.1.min.js" integrity="sha256-hwg4gsxgFZhOsEEamDoyGBf13FyQuiTw1AQgxVSNgt4=" crossorigin="anonymous"></script>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo" crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnuUD4Ih7YwaYd1iqfktj0Uod8GCEx13Og8ifwB6" crossorigin="anonymous"></script>
```

```
<script type="module" src="js/main.js"></script>

</body>
</html>
```

### Sourcecode - style.css:

```
body {
    font-family: 'Raleway', sans-serif;
}

p {
    text-align: center;
}

h1, h2 {
    text-align: center;
}

/**************************************** NAVIGATION *****/
nav {
```

```
background: rgb(168, 229, 198);
padding: 1em;
}

nav a {
    color: white;
    text-decoration: none;
    margin: 0.5em;
    font-size: 20px;
}

nav a:hover {
    text-decoration: overline;
}

/****** RESPONSIVE NAV *****/
@media (max-device-width: 720px) {
    nav a{
        display: block;
    }
}

/****** LINKS *****/
a {
    text-decoration: none;
}

a:hover {
    color: white;
    background-color: rgb(168, 229, 198);
    text-decoration: overline;
}

/****** TEXT ALIGN *****/
.left {
    text-align: left;
    margin: 1em;
}
```

```
.left_small {  
    text-align: left;  
    margin-left: 1em;  
    font-size: 14px;  
}  
  
.left_small_one {  
    text-align: left;  
    margin-left: 1em;  
    margin-bottom: -1em;  
    font-size: 14px;  
}  
  
/****** LIST *****/  
.list{  
    display: flex;  
    justify-content: space-around;  
    flex-wrap: wrap;  
    align-content: space-around;  
    background-image: url("assets/foodism360-GfuxfFJt7cw-unplash.jpg");  
}  
  
.list div{  
    width: 250px;  
    border: solid rgb(168, 229, 198) 2px;  
    margin: 10px;  
    text-align: center;  
    background-color: white;  
}  
  
/****** BUTTON *****/  
button{  
    background-color: rgb(168, 229, 198);  
    margin-bottom: 1em;  
    border: 0px;  
    padding: 0.7em;
```

```
border-radius: 2em;
margin-right: 1em;
}

.detail{
    font-family: 'Raleway', sans-serif;
    color: rgb(168, 229, 198);
    background-color: white;
    font-size: 1.2em;
    padding: 0em;
    margin: 0.2em;
}

.detail:hover{
    color: #4d172e;
}

/****** RESPONSIVE TABLE *****/

@media (max-width: 720px) {
    #detail {
        grid-template-columns: repeat(3,1fr);
    }
}

/****** TABLE *****/

table, tr, td{
    border-bottom: 1px solid black;
    color: black;
}

table {
    margin: 5em;
}

th{
    background-color: rgb(168, 229, 198);
    font-size: 20px;
```

```
text-align: left;
display: grid;
padding: 1%;
grid-template-columns: repeat(3, 1fr);
grid-template-rows: repeat(1, 1fr);
}

tr{
    display: grid;
    grid-template-columns: repeat(3 ,1fr);
    grid-template-rows: repeat(1, 1fr);
}

td{
    padding: 3%;
    display: grid;
}

/***** FLEX *****/
.middle {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
}

.button, #btn-help, #newList, #back{
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
}

.topbuttons{
    border: 1px solid grey;
    background-color: white;
    border-radius: 0.5em;
    margin-top: 2em;
}
```

```
    transition: background-color .5s ease-in;
}

.topbuttons:hover{
    background-color: rgba(70, 0, 96, 0.82);
    color: white;
}

withoutBorder{
    border: none;
    background: none;
}

/***** STATUS *****/
.list .offen {
    border-left: 5px solid rgb(100, 229, 125);
}

.list .geschlossen {
    border-left: 5px solid rgb(229, 109, 110);
}

.hide{
    display: none;
}

.list .grey{
    color: grey;
}

.article{
    font-family: 'Raleway', sans-serif;
    color: rgb(168, 229, 198);
    background-color: white;
```

```
font-size: 1.2em;  
padding: 0em;  
margin: 0.2em;  
}
```

### Sourcecode - lists.json:

```
{  
  "listId": 0,  
  "lists": [  
    {  
      "id": 0,  
      "title": "Hofer",  
      "name": "Waltraud Binder",  
      "description": "Diese Artikel brauche ich von Hofer.",  
      "date": "11.12.2020",  
      "dueDate": "20.12.2020",  
      "articleNr": "5",  
      "article": [  
        {  
          "quantity": "500g",  
          "name": "Nudeln",  
          "cost": "3"  
        },  
        {  
          "quantity": "2 Gläser",  
          "name": "Pesto",  
          "cost": "5"  
        },  
        {  
          "quantity": "2 L",  
          "name": "Olivenöl",  
          "cost": "8.50"  
        },  
        {  
          "quantity": "6 Stk",  
          "name": "Crossiants",  
          "cost": "4.50"  
        },  
      ]  
    }  
  ]  
}
```

```
{  
    "quantity": "2 Stk",  
    "name": "Gurken",  
    "cost": "3"  
}  
]  
,  
{  
    "id": 1,  
    "title": "Merkur",  
    "name": "Franz Herzog",  
    "description": "Diese Artikel brauche ich von Merkur.",  
    "date": "15.12.2020",  
    "dueDate": "3.1.2021",  
    "articleNr": "8",  
    "article": [  
        {  
            "quantity": "500g",  
            "name": "Nudeln",  
            "cost": "3"  
        },  
        {  
            "quantity": "1 Packung",  
            "name": "Haribo",  
            "cost": "5"  
        },  
        {  
            "quantity": "3",  
            "name": "Paprika",  
            "cost": "5"  
        },  
        {  
            "quantity": "1 Packung",  
            "name": "Toast",  
            "cost": "5"  
        },  
        {  
            "quantity": "2 Gläser",  
            "name": "Milch",  
            "cost": "2"  
        }  
    ]  
}
```

```
"name": "Tomatensauce",
"cost": "5"
},
{
"quantity": "2 L",
"name": "Olivenöl",
"cost": "8.50"
},
{
"quantity": "6 Stk",
"name": "Crossiants",
"cost": "4.50"
},
{
"quantity": "2 Stk",
"name": "Gurken",
"cost": "50"
}
]
},
{
"id": 2,
"title": "Libro",
"name": "Hugo Kathan",
"description": "Diese Artikel brauche ich von Libro.",
"date": "13.1.2021",
"dueDate": "5.2.2021",
"articleNr": "3",
"article": [
{
"quantity": "3",
"name": "Kugelschreiber",
"cost": "5"
},
{
"quantity": "1",
"name": "Zeichenblock",
"cost": "6"
}
```

```
        },
        {
            "quantity": "1 Packung",
            "name": "Buntstifte",
            "cost": "10"
        }
    ],
},
{
    "id": 3,
    "title": "DM",
    "name": "Lotte Siegmann",
    "description": "Diese Artikel brauche ich von DM.",
    "date": "27.12.2020",
    "dueDate": "4.1.2021",
    "articleNr": "5",
    "article": [
        {
            "quantity": "1",
            "name": "Wimperntusche",
            "cost": "10"
        },
        {
            "quantity": "3",
            "name": "Tuchmasken",
            "cost": "9"
        },
        {
            "quantity": "2",
            "name": "Nagellack Rot und Gelb",
            "cost": "5"
        },
        {
            "quantity": "1",
            "name": "Bebe Gesichtscreme",
            "cost": "6"
        },
        {

```

```
        "quantity": "1",
        "name": "Bürste",
        "cost": "11"

    }
]

},
{
    "id": 4,
    "title": "Thalia",
    "name": "Silvia Gümherr",
    "description": "Diese Artikel brauche ich von Thalia.",
    "date": "9.12.2020",
    "dueDate": "12.1.2021",
    "articleNr": "2",
    "article": [
        {
            "quantity": "1",
            "name": "Harry Potter - Stein der Waisen",
            "cost": "20"
        },
        {
            "quantity": "1",
            "name": "Känguru Chroniken Teil 1",
            "cost": "23"
        }
    ]
},
"userId": 0,
"user": [
    {
        "id": 1,
        "firstName": "Hugo",
        "lastName": "Kathan",
        "username": "huka"
    }
]
}
```

### Sourcecode - main.js:

```
import SocialShopping from "./socialShopping.js";
//creates a new map
let covidSocialShopping = new SocialShopping();
$(document).ready(function () {
    //initialises the methods loadFromJSON and addEventHandlers
    covidSocialShopping.init();
});
```

### Sourcecode - SocialShopping.js:

```
import List from "./list.js";
import ShoppingList from "./shoppingList.js";
import Article from "./article.js";
import User from "./user.js";

let addEventHandler = Symbol();
let loadFromJSON = Symbol();
let addArticleToList = Symbol();

export default class SocialShopping {
    #shoppingList;
    #listID;
    #user;

    constructor() {
        this.#shoppingList = new ShoppingList();
    }

    //initialise the loadFromJSON and addEventHandler functions, to use it in
    main.js
    init() {
        this[loadFromJSON]();
        this[addEventHandler]();
    }
}
```

```
}
```

```
//add event handlers
[addEventListener] () {

    //HELPER VIEW
    //Show functions the helper has and hide functions he does not have
    $("#btn-help").on("click", "#helper", function () {
        $(".delete").hide();
        $(".edit").hide();
        $(".take-list").show();
        $(".confirm").show();
        $(".input").show();
        $("#newList").hide();
    });

    //SEEKING HELP VIEW
    //Show functions the person has who is looking for help, hide other
    //functions
    $("#btn-help").on("click", "#help", function () {
        $(".delete").show();
        $(".edit").show();
        $(".take-list").hide();
        $(".confirm").hide();
        $(".input").hide();
        $("#newList").show();
    });

    //GO BACK
    //Go back to list view
    $("#back").on("click", "#goBack", function () {
        $("#detailview").addClass("hide");
        $(".list").removeClass("hide");
        $("#details").empty();
        $(".tbody").empty();
    });
}

let newList;
```

```

let creationDate;
let that = this;

//NEW LIST
//Make a new list by clicking on "Neue Liste hinzufügen"
//Showing of pop-up form where user can enter list details and articles
//creation date is already given (today's date)

$("#newList").on("click", "#addList", function () {
    creationDate = new Date().toLocaleDateString();
    newList = new Map();

    $("#dateCreationAdd").html(creationDate);
});

//Add articles by clicking on "Artikel hinzufügen"
//articles are added to the map and also shown at the bottom

$("form").on("click", "#addArticleToList", function () {
    let name = $("#ArticleAdd").val();
    let quantity = $("#ArticleAddQuantity").val();
    let cost = $("#ArticleAddPrice").val();

    let article = new Article({name, quantity, cost});
    newList.set(name, article);

    $("#forArticles").append(`<div>
        <div>Artikel: ${name}</div>
        <div>Menge: ${quantity}</div>
        <div>max. Kosten: ${cost}</div>
    </div>`);

    $("#ArticleAdd").val("");
    $("#ArticleAddQuantity").val("");
    $("#ArticleAddPrice").val("");
});

//Make a new list by clicking on "Liste hinzufügen"
//Shows list with the input

```

```

$( ".modal-footer" ).on( "click", "#listAdd", function () {
    let length = $( ".list" ).length + 1;
    let listId = "list" + length;
    let name = $( "#nameCreatorAdd" ).val();
    let description = $( "#descriptionAdd" ).val();
    let title = $( "#titleListAdd" ).val();
    let dueDate = $( "#dueDateAdd" ).val();

    let list = new List({name, listId, title, description, creationDate,
    dueDate});

    list.useArticles(newArticleList);
    that.#shoppingList.addList(list);

    list.printList();
    $("form input").val('');
    $("#forArticles").empty();

}) ;

//TAKE LIST
//Sets the status to open, because the list is taken and has to be
finished
$(".list").on("click", ".take-list", function () {
    $(this).parent().addClass("offen");
}) ;

//DETAIL VIEW
//Shows the articles in the detail view
$(".list").on("click", ".detail", function () {

    $("#detailview").removeClass("hide");
    $(".list").addClass("hide");

    let id = $(this.parentElement.parentElement).attr("id");
}

```

```

        let a = that.#shoppingList.getListById(Number(id));
        a.printAllArticles();
        a.printDetails();
    });

//DELETE
//Deletes lists, also from map after confirming that the list should be deleted
$(".list").on("click", ".delete", function () {
    let list = $(this).parent();
    if (confirm('Willst du diese Liste wirklich löschen?')) {
        let id = $(this.parentElement).attr("id");
        let listId = that.#shoppingList.getListById(Number(id));
        that.#shoppingList.deleteList(listId);
        list.remove();
    }
});

//EDIT LIST
$(".list").on("click", ".edit", function () {
    if (confirm('Willst du diese Liste wirklich bearbeiten?')) {
        // make the span editable and focus it
        $(this).closest("div").find("p").prop("contenteditable", true).focus();
        return false;
    }
});

//SET DONE THROUGH PRICE
//After a total price is added list is set to the status: done and disabled
$(".list").on("click", ".confirm", function () {
    $(this).parent().addClass("geschlossen");
    $(this).siblings('button').remove();
    $(this).siblings().children('button').remove();
    $(this).siblings().children('input').prop('disabled', true);
});

```

```
$($this).remove();  
});  
  
//loads the map from JSON to be able to show the entered values for the  
various lists  
[loadFromJSON] () {  
  
    $.getJSON("json/lists.json").then((data)=> {  
        console.log(data);  
  
        this.#listID = data.listId;  
  
        for(let lists of data.lists) {  
  
            let l = new List(lists);  
            this.#shoppingList.addList(l);  
            console.log(this.#shoppingList);  
            this[addArticleToList](l, lists);  
  
        }  
  
        this.#shoppingList.printShoppingList();  
  
    });  
}  
  
//adds the articles to the list  
[addArticleToList](list, jsonList) {  
    for (let a of jsonList.article){  
        let article = new Article(a);  
        list.addArticle(article);  
    }  
}  
}
```

### Sourcecode - ShoppingList.js:

```

export default class ShoppingList {

    #shoppingList;

    constructor() {
        this.#shoppingList = new Map();
    }

    //adds a list
    addList(list) {
        this.#shoppingList.set(list.id, list);
    }

    //Get a list by its ID
    getListById(listID) {
        return this.#shoppingList.get(listID);
    }

    //prints all lists
    printShoppingList() {
        for (let list of this.#shoppingList.values()) {
            list.printList();
        }
    }

    //deletes the lists from the map
    deleteList(listID) {
        this.#shoppingList.delete(listID);
    }
}

```

### Sourcecode - list.js:

```

export default class List {

    #id;
    #title;
    #name;
    #description;
    #dueDate;
    #date;
    #articleNr;
    #article;
}

```

```
constructor({id, title, name, description, date, dueDate, articleNr}) {  
    this.#id = id;  
    this.#title = title;  
    this.#name = name;  
    this.#description = description;  
    this.#date = date;  
    this.#dueDate = dueDate;  
    this.#articleNr = articleNr;  
    this.#article = [];  
}  
  
//Get ID of List  
get id() {  
    return this.#id;  
}  
  
//Get title of List  
get title() {  
    return this.#title;  
}  
  
//Get name of List  
get name() {  
    return this.#name;  
}  
  
//Get description of List  
get description() {  
    return this.#description;  
}  
  
//Get date of List  
get date() {  
    return this.#dueDate;  
}  
  
//Get numbers of Articles in List
```

```

get articleNr() {
    return this.#articleNr;
}

//Add an Article to the list
addArticle(a) {
    this.#article.push(a);
}

//Get the numbers of articles from a newly added list
useArticles(articleMap) {
    this.#article = articleMap;
    this.#articleNr = articleMap.size;
}

//Print a list with an overview (important characteristics, buttons and
description)
printList() {

    let id = this.#id;
    let title = this.#title;
    let name = this.#name;
    let desc = this.#description;
    let dueDate = this.#dueDate;
    let anum = this.#articleNr;

    let listElement = $(`

Christina Wanke



24


```

```

        <p class="input">Gesamtpreis: <input type="text" id="input" size="5">€</p>

        <button class="confirm">Bestätigen</button>
    </div>) ;

//Append list after the div with the class: list
$(".list").append(listElement);

}

//Go through all articles in a list and print one after the other,
//in the end all articles are printed
printAllArticles() {
    for (let a of this.#article) {
        a.printArticle();
    }
}

//Prints the details of thr list into the detailview
printDetails() {
    let name = this.#name;
    let date = this.#date;
    let desc = this.#description;
    let dueDate = this.#dueDate;
    let anum = this.#articleNr;

    $("#detail").addClass(this.#id);

    let detail = $(`

## Einkaufsliste von ${name}</h2> ${anum} Artikel</p> Erstellt am: ${date}</p> Zu erfüllen am: ${dueDate}</p> ${desc}</p> `); $("#details").prepend(detail); } }


```

## Sourcecode - article.js:

```
export default class Article {  
    #quantity;  
    #name;  
    #cost;  
  
    constructor({quantity, name, cost}) {  
        this.#quantity = quantity;  
        this.#name = name;  
        this.#cost = cost;  
    }  
  
    //Get name of article  
    get name() {  
        return this.#name;  
    }  
  
    //Set a new name for article  
    set name(newName) {  
        this.#name = newName;  
    }  
  
    //Get quantity of article  
    get quantity() {  
        return this.#quantity;  
    }  
  
    //Set a new quantity for article  
    set quantity(newQuantity) {  
        this.#quantity = newQuantity;  
    }  
  
    //Get cost of article  
    get cost() {  
        return this.#cost;  
    }  
  
    //Set a new cost for article
```

```

set cost(newCost) {
    this.#cost = newCost;
}

//print an article into the table from the detailview
printArticle() {

    let name = this.#name;
    let quant = this.#quantity;
    let cost = this.#cost;

    let articles = `

        <tr class="article">
            <td>${quant}</td>
            <td>${name}</td>
            <td>${cost}</td>
        </tr>`;
}

//append the article to the table body
$(".tablebody").append(articles);
}
}

```

### Sourcecode - user.js:

```

export default class User {

    #id;
    #firstName;
    #lastName;
    #username;

    constructor(id, firstName, lastName, username) {
        this.#id = id;
        this.#firstName = firstName;
        this.#lastName = lastName;
        this.#username = username;
    }
}

```

## Testfälle:

View des Hilfesuchenden:

The screenshot shows a web browser window for 'COVID Social Shopping' at localhost:63342. The top navigation bar includes links for Home, Einkaufslisten, Chat, Anmelden, and Registrieren. Below the navigation is a header with two tabs: 'Hilfesuchender' (selected) and 'Helfender'. The main title is 'COVID Social Shopping'. A button 'Neue Liste hinzufügen' is visible. The page displays five shopping lists (Hofer, Merkur, Libro, DM, Thalia) each with a summary of required items and a 'bearbeiten' or 'löschen' button.

Shop	Anzahl Artikel	Erstellt von
Hofer	5 Artikel	Waltraud Binder
Merkur	8 Artikel	Franz Herzog
Libro	3 Artikel	Hugo Kathan
DM	5 Artikel	Lotte Siegmann
Thalia	2 Artikel	Silvia Gümherr

View des Helfenden:

The screenshot shows the same web browser window for 'COVID Social Shopping' at localhost:63342. The 'Helfender' tab is selected in the header. The main title is 'COVID Social Shopping'. The five shopping lists (Hofer, Merkur, Libro, DM, Thalia) now include a 'übernehmen' button instead of 'bearbeiten' or 'löschen'. Each list also includes a 'Gesamtpreis:' field and a 'Bestätigen' button.

Shop	Anzahl Artikel	Erstellt von	Gesamtpreis:	Aktion
Hofer	5 Artikel	Waltraud Binder	<input type="text"/> €	übernehmen Bestätigen
Merkur	8 Artikel	Franz Herzog	<input type="text"/> €	übernehmen Bestätigen
Libro	3 Artikel	Hugo Kathan	<input type="text"/> €	übernehmen Bestätigen
DM	5 Artikel	Lotte Siegmann	<input type="text"/> €	übernehmen Bestätigen
Thalia	2 Artikel	Silvia Gümherr	<input type="text"/> €	übernehmen Bestätigen

Der Hilfesuchende erstellt eine neue Liste:

The screenshot shows a web browser window for 'COVID Social Shopping' at the URL [localhost:63342/UE01\\_COVID\\_Social\\_Shopping/index.html?\\_jst=df787mfe04afmf8mj8bmormdi#](http://localhost:63342/UE01_COVID_Social_Shopping/index.html?_jst=df787mfe04afmf8mj8bmormdi#). A modal dialog box titled 'Einkaufsliste hinzufügen' is open. It contains fields for 'Name des Erstellers' (Lotte Rieger), 'Erstellungsdatum' (20.1.2021), 'Titel der Einkaufsliste' (Penny), 'Beschreibung' (Diese Artikel brauche ich von Penny), 'Gewünschtes Erfüllungsdatum' (20.3.2021), 'Artikel' (Artikelbezeichnung), 'Menge' (Menge), and 'maximaler Preis'. Below the form, there is a link 'Artikel hinzufügen' and a preview of the added item: Artikel: Tomaten, Menge: 2, max. Kosten: 3, Artikel: Nudeln. In the background, there are four shopping lists: Hofer (5 Artikel), Merkur (8 Artikel), DM (5 Artikel), and Thalia (2 Artikel). Each list has a 'bearbeiten' and 'löschen' button.

Der Hilfesuchende klickt auf Artikel hinzufügen:

The screenshot shows the same web browser window and modal dialog as the previous one. The 'Artikel hinzufügen' link is now highlighted in green. The preview below the form shows the added item: Artikel: Tomaten, Menge: 2, max. Kosten: 3, Artikel: Nudeln. A small thumbnail image of the application interface is visible in the bottom right corner of the main window.

Der Hilfesuchende fügt die Liste hinzu:

The screenshot shows a web browser window for 'COVID Social Shopping' at the URL `localhost:63342/UE01_COVID_Social_Shopping/index.html?_j= df787mf04afmf8rnj8bmormdi#`. The page displays several shopping lists from different stores: Hofer, Merkur, Libro, DM, and Thalia. A new list, 'Penny', is being added. The form fields for the new list include:

- Erstellungsdatum: 20.1.2021
- Titel der Einkaufsliste: Penny
- Beschreibung: Diese Artikel brauche ich von Penny
- Gewünschtes Erfüllungsdatum: 20.3.2021
- Artikel: Artikelbezeichnung
- Menge
- maximaler Preis

Below the form, there is a list of articles with their details:

- Artikel: Tomaten  
Menge: 2  
max. Kosten: 3
- Artikel: Nudeln  
Menge: 500g  
max. Kosten: 3
- Artikel: Mozarella  
Menge: 3  
max. Kosten: 5

At the bottom right of the main content area, there is a small thumbnail image of a mobile device displaying the application's interface.

Die Liste wurde hinzugefügt:

The screenshot shows the same web browser window for 'COVID Social Shopping' at the URL `localhost:63342/UE01_COVID_Social_Shopping/index.html?_j= df787mf04afmf8rnj8bmormdi#`. The page now includes the newly added 'Penny' list, which is highlighted with a green border. The other lists (Hofer, Merkur, Libro, DM, Thalia) are also visible with their respective article counts and user information.

## Detailview:

The screenshot shows a web browser window with the title "COVID Social Shopping". The URL is "localhost:63342/UE01\_COVID\_Social\_Shopping/index.html?\_ijt=df787mfeo4afmf8nj8bmormdi#". The main content is titled "Einkaufsliste von Waltraud Binder" and displays a table of items with columns: Anzahl, Titel, and max. Kosten in €.

Anzahl	Titel	max. Kosten in €
500g	Nudeln	3
2 Gläser	Pesto	5
2 L	Olivenöl	8.50
6 Stk	Crossiants	4.50
2 Stk	Gurken	3

Below the table are two buttons: "Zurück" and "Bearbeiten". A small preview of the list is shown in a rounded rectangle on the right.

## Liste bearbeiten:

The screenshot shows a web browser window with the title "COVID Social Shopping". A modal dialog box is open, asking "Willst du diese Liste wirklich bearbeiten?" with "Abbrechen" and "Ok" buttons. The background shows a grid of five shopping lists for different users: Hofer, Merkur, Libro, DM, and Thalia.

Hofer	Merkur	Libro	DM	Thalia
Diese Artikel brauche ich von Hofer. Erfüllungsdatum: 20.12.2020  5 Artikel  <a href="#">bearbeiten</a> <a href="#">löschen</a> Liste von: Waltraud Binder	Diese Artikel brauche ich von Merkur. Erfüllungsdatum: 3.1.2021  8 Artikel  <a href="#">bearbeiten</a> <a href="#">löschen</a> Liste von: Franz Herzog	Diese Artikel brauche ich von Libro. Erfüllungsdatum: 5.2.2021  3 Artikel  <a href="#">bearbeiten</a> <a href="#">löschen</a> Liste von: Hugo Kathan	Diese Artikel brauche ich von DM. Erfüllungsdatum: 4.1.2021  5 Artikel  <a href="#">bearbeiten</a> <a href="#">löschen</a> Liste von: Lotte Siegmann	Diese Artikel brauche ich von Thalia. Erfüllungsdatum: 12.1.2021  2 Artikel  <a href="#">bearbeiten</a> <a href="#">löschen</a> Liste von: Silvia Gümherr

The screenshot shows the main interface of the COVID Social Shopping application. At the top, there is a navigation bar with links for Home, Einkaufslisten, Chat, Anmelden, and Registrieren. Below the navigation bar, there are two buttons: "Hilfesuchender" and "Helfender". The main title "COVID Social Shopping" is centered above a button labeled "Neue Liste hinzufügen". Below the title, there are five cards representing different shopping lists:

- Hofer**: Contains text about needing items from Hofer for Christmas, an fulfillment date of 20.12.2020, and 5 articles. Buttons for "bearbeiten" and "löschen" are present.
- Merkur**: Contains text about needing items from Merkur, an fulfillment date of 31.2021, and 8 articles. Buttons for "bearbeiten" and "löschen" are present.
- Libro**: Contains text about needing items from Libro, an fulfillment date of 5.2.2021, and 3 articles. Buttons for "bearbeiten" and "löschen" are present.
- DM**: Contains text about needing items from DM, an fulfillment date of 4.1.2021, and 5 articles. Buttons for "bearbeiten" and "löschen" are present.
- Thalia**: Contains text about needing items from Thalia, an fulfillment date of 12.1.2021, and 2 articles. Buttons for "bearbeiten" and "löschen" are present.

### Liste löschen:

The screenshot shows the same interface as the previous one, but with a modal dialog box in the center asking for confirmation to delete the Merkur list. The dialog contains the text "Auf localhost:63342 wird Folgendes angezeigt: Willst du diese Liste wirklich löschen?" with "Abbrechen" and "Ok" buttons.

Below the dialog, the main content area displays the five shopping lists again. The Merkur list is still visible but appears slightly faded or less prominent than the others.

The screenshot shows the application interface from the perspective of a helper. At the top, there are two buttons: "Hilfesuchender" (Helper) and "Helfender" (Helper). Below them is the title "COVID Social Shopping". A button "Neue Liste hinzufügen" (Add new list) is visible. The main area displays four requests from users:

- Hofer**: Diese Artikel brauche ich von Hofer. Es ist sehr dringend, ich brauche sie für Weihnachten. Erfüllungsdatum: 20.12.2020. 5 Artikel. Buttons: bearbeiten, löschen. Listed by Waltraud Binder.
- Libro**: Diese Artikel brauche ich von Libro. Erfüllungsdatum: 5.2.2021. 3 Artikel. Buttons: bearbeiten, löschen. Listed by Hugo Kathan.
- DM**: Diese Artikel brauche ich von DM. Erfüllungsdatum: 4.1.2021. 5 Artikel. Buttons: bearbeiten, löschen. Listed by Lotte Siegmann.
- Thalia**: Diese Artikel brauche ich von Thalia. Erfüllungsdatum: 12.1.2021. 2 Artikel. Buttons: bearbeiten, löschen. Listed by Silvia Gümherr.

A small inset image in the bottom right corner shows the application from the perspective of a requester (Hilfesuchender).

## Sicht des Helfenden:

The screenshot shows the application interface from the perspective of a helper. At the top, there are two buttons: "Hilfesuchender" (Helper) and "Helfender" (Helper), with "Helfender" being the active tab. Below them is the title "COVID Social Shopping". The main area displays five lists from users:

- Hofer**: Diese Artikel brauche ich von Hofer. Erfüllungsdatum: 20.12.2020. 5 Artikel. Buttons: übernehmen, Bestätigen. Listed by Waltraud Binder. Gesamtpreis: [ ] €.
- Merkur**: Diese Artikel brauche ich von Merkur. Erfüllungsdatum: 3.1.2021. 8 Artikel. Buttons: übernehmen, Bestätigen. Listed by Franz Herzog. Gesamtpreis: [ ] €.
- Libro**: Diese Artikel brauche ich von Libro. Erfüllungsdatum: 5.2.2021. 3 Artikel. Buttons: übernehmen, Bestätigen. Listed by Hugo Kathan. Gesamtpreis: [ ] €.
- DM**: Diese Artikel brauche ich von DM. Erfüllungsdatum: 4.1.2021. 5 Artikel. Buttons: übernehmen, Bestätigen. Listed by Lotte Siegmann. Gesamtpreis: [ ] €.
- Thalia**: Diese Artikel brauche ich von Thalia. Erfüllungsdatum: 12.1.2021. 2 Artikel. Buttons: übernehmen, Bestätigen. Listed by Silvia Gümherr. Gesamtpreis: [ ] €.

Liste übernehmen und Status auf „offen“ setzen:

The screenshot shows the COVID Social Shopping application interface. At the top, there are two buttons: "Hilfesuchender" and "Helfender". Below them, the title "COVID Social Shopping" is centered. There are four separate boxes, each representing a different user's shopping list:

- Hofer:** Text: "Diese Artikel brauche ich von Hofer. Es ist sehr dringend, ich brauche sie für Weihnachten." Date: "Erfüllungsdatum: 20.12.2020". Buttons: "5 Artikel" (green), "übernehmen" (green), "Bestätigen" (green).
- Libro:** Text: "Diese Artikel brauche ich von Libro." Date: "Erfüllungsdatum: 5.2.2021". Buttons: "3 Artikel" (green), "übernehmen" (green), "Bestätigen" (green).
- DM:** Text: "Diese Artikel brauche ich von DM." Date: "Erfüllungsdatum: 4.1.2021". Buttons: "5 Artikel" (green), "übernehmen" (green), "Bestätigen" (green).
- Thalia:** Text: "Diese Artikel brauche ich von Thalia." Date: "Erfüllungsdatum: 12.1.2021". Buttons: "2 Artikel" (green), "übernehmen" (green), "Bestätigen" (green).

Gesamtpreis eingeben:

The screenshot shows the same COVID Social Shopping application interface as the previous one, but with a change in the Hofer request. The "Gesamtpreis" field for the Hofer request now contains the value "23 €". The other requests remain unchanged.

Gesamtpreis bestätigen, Liste auf „erledigt“ setzen und sperren:

The screenshot shows a web browser window for 'COVID Social Shopping' at the URL 'localhost:63342/UE01\_COVID\_Social\_Shopping/index.html'. The page has a green header bar with links for 'Home', 'Einkaufslisten', 'Chat', 'Anmelden', and 'Registrieren'. Below the header are two buttons: 'Hilfesuchender' and 'Helfender'. The main content area displays four separate shopping lists:

- Hofer**: Text: "Diese Artikel brauche ich von Hofer. Es ist sehr dringen, ich brauche sie für Weihnachten." Date: "Erfüllungsdatum: 20.12.2020". Author: "Liste von: Waltraud Binder". Total price: "Gesamtpreis: 23 €". Buttons: 'Bestätigen' (green) and 'Übernehmen' (green).
- Libro**: Text: "Diese Artikel brauche ich von Libro." Date: "Erfüllungsdatum: 5.2.2021". Author: "Liste von: Hugo Kathan". Total price: "Gesamtpreis: [ ] €". Buttons: 'Bestätigen' (green) and 'Übernehmen' (green).
- DM**: Text: "Diese Artikel brauche ich von DM." Date: "Erfüllungsdatum: 4.1.2021". Author: "Liste von: Lotte Siegmann". Total price: "Gesamtpreis: [ ] €". Buttons: 'Bestätigen' (green) and 'Übernehmen' (green).
- Thalia**: Text: "Diese Artikel brauche ich von Thalia." Date: "Erfüllungsdatum: 12.1.2021". Author: "Liste von: Silvia Gümherr". Total price: "Gesamtpreis: [ ] €". Buttons: 'Bestätigen' (green) and 'Übernehmen' (green).

A small icon in the bottom right corner shows a mobile device displaying the app's interface.