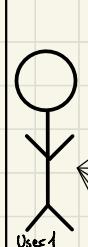


Use-case Diagramm:

Client:



Start

Product

Edit Product

Delete Product

Check Product

request
response

create Name
create Amount
create Comment

Add Product

edit Amount
edit Comment
edit Importance

remove all attributes

create date
of purchase

Server:

Load data

Update List

send
response
send
response
send
response

Storage:

findList

User Interface

Einkaufsliste

Produkt	Anzahl	Kommentar	Hinzufügen
<input checked="" type="checkbox"/> Kellogs Smacks	1	01.10.2022 XXL Packung	 
<input type="checkbox"/> Fruchtzwerge	2	15.10.2022 Packung	 

```
<div>  
id = addToList
```

```
<input>  
type = text  
class = boxInput  
placeholder = "Produkt"
```

```
<input>  
type = number  
class = boxInput  
placeholder = "Anzahl"
```

```
<div>  
id = item0  
class = buy
```

```
<div>  
id = item1  
class = buyNext
```

```
<input>  
type = checkbox  
id = check0  
> click
```

```
<h1>
```

```
<input>  
type = text  
class = boxInput  
placeholder = "Kommentar"
```

```
<button>  
id = add  
class = boxInput  
> click
```

```
<i>  
id = trash0  
> click  
class = fas fa-trash
```

```
<i>  
id = edit0  
> click  
class = fas fa-pen
```

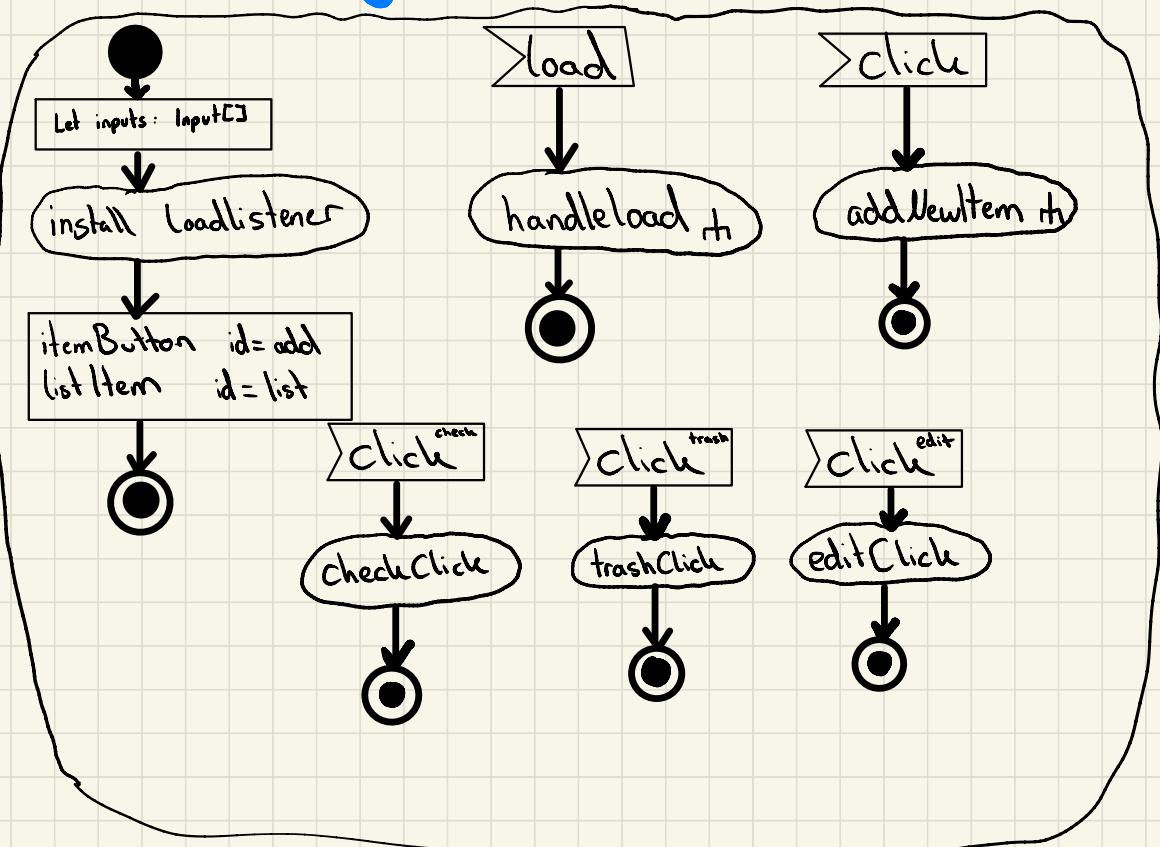
```
<p>  
class = product
```

```
<p>  
class = amount
```

```
<p>  
class = date
```

```
<p>  
class = comment
```

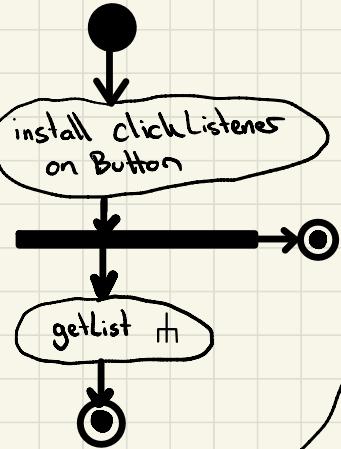
Aktivitätsdiagramm



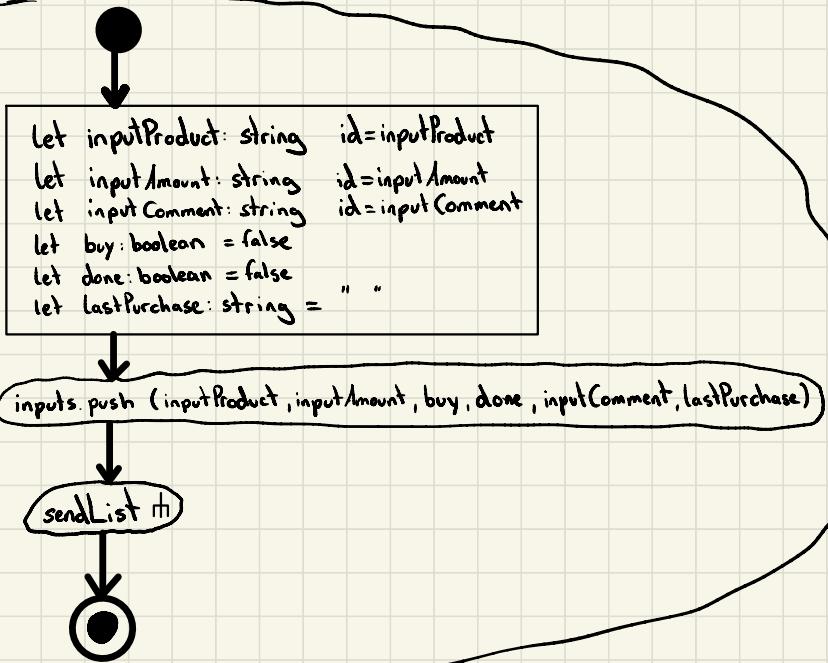
interface Input

```
interface Input {  
    inputProduct: string;  
    inputAmount: number;  
    buy: boolean;  
    done: boolean;  
    inputComment: string;  
    lastPurchase: string;  
}
```

handleLoad



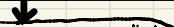
addNewItem



writeItems



```
let list = Div with id=list
```



```
list.innerHTML = ""
```



```
let index: number = 0
```



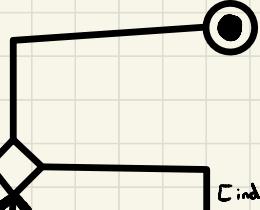
```
[index < inputs.length]
```

```
create div with id = index  
class = buy
```

```
create checkbox with id = checkbox + index  
create p with id = item + index  
create button with id = trash + index  
class = fas fa-trash  
create button with id = edit + index  
class = fas fa-pen
```

```
listItem.appendChild (newDiv)
```

```
newDiv.appendChild (checkbox, p,  
button, button)
```



```
[index < inputs.length]
```

```
let item = document (#item + index)  
let checkBox = document (#checkbox + index)  
let trashButton = document (#trash + index)  
let editButton = document (#edit + index)
```

```
item.innerHTML = inputs[index].inputProduct + " " +  
inputs[index].inputAmount + " " +  
inputs[index].inputComment + " " +  
inputs[index].lastPurchase
```

```
install click listeners on  
checkbox, trash- and editButton
```

trashClick

_event: MouseEvent

```
let id = target of _event
let newId: number = cutID(id, 5)
```

inputs.splice(newId, 1)

sendList ()

cutID

_id: string, _length: number

```
let newId: string = _id.slice(_length)
```

```
return parseInt(newId)
```

check Click

_event: MouseEvent

```
let id: string = (_event.target).id
let newId: number = cutID(id, 8)
let date: Date = new Date()
let day: number = date.getDate()
let month: number = (new Date().getMonth() + 1)
let year: number = date.getFullYear()
```

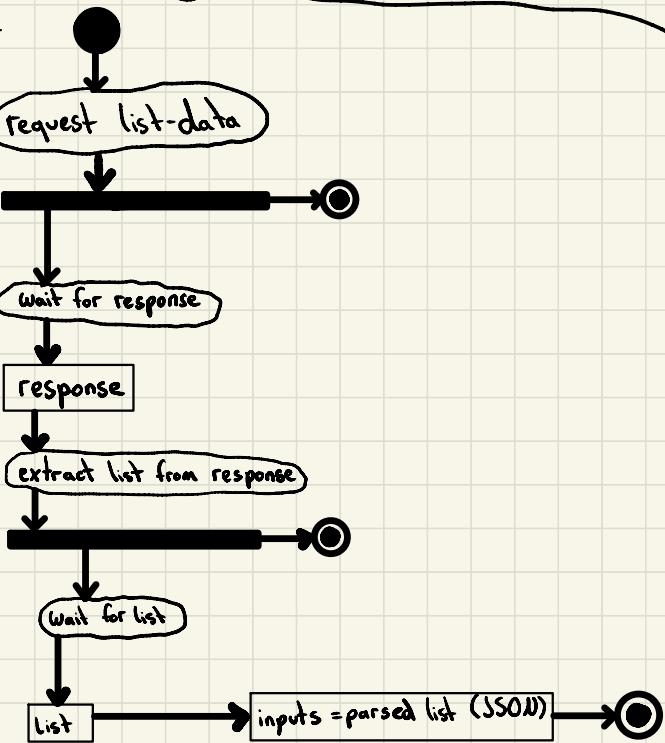
inputs[newId].lastPurchase = day + " " + month + " " + year

sendList ()

edit Click



getList



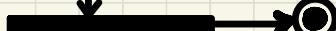
SendList



```
(let sendInputs: string  
= JSON.stringify(inputs))
```

```
query = create query  
from sendInputs
```

send query to server



Wait for response

notify user about
request send

