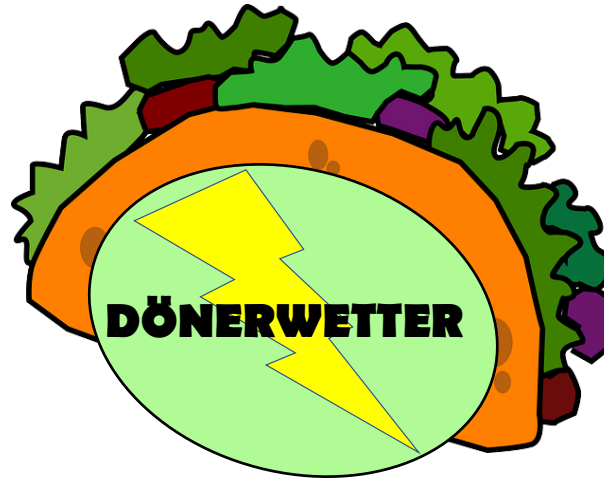


# KONZEPT

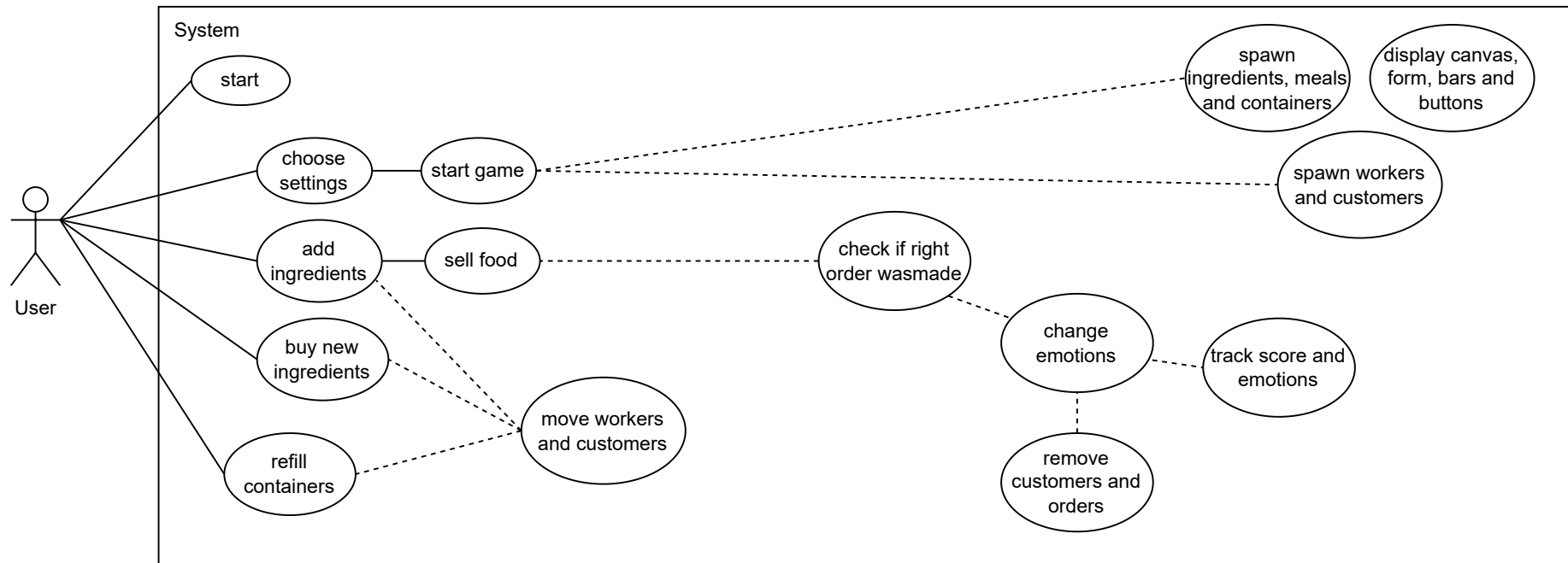


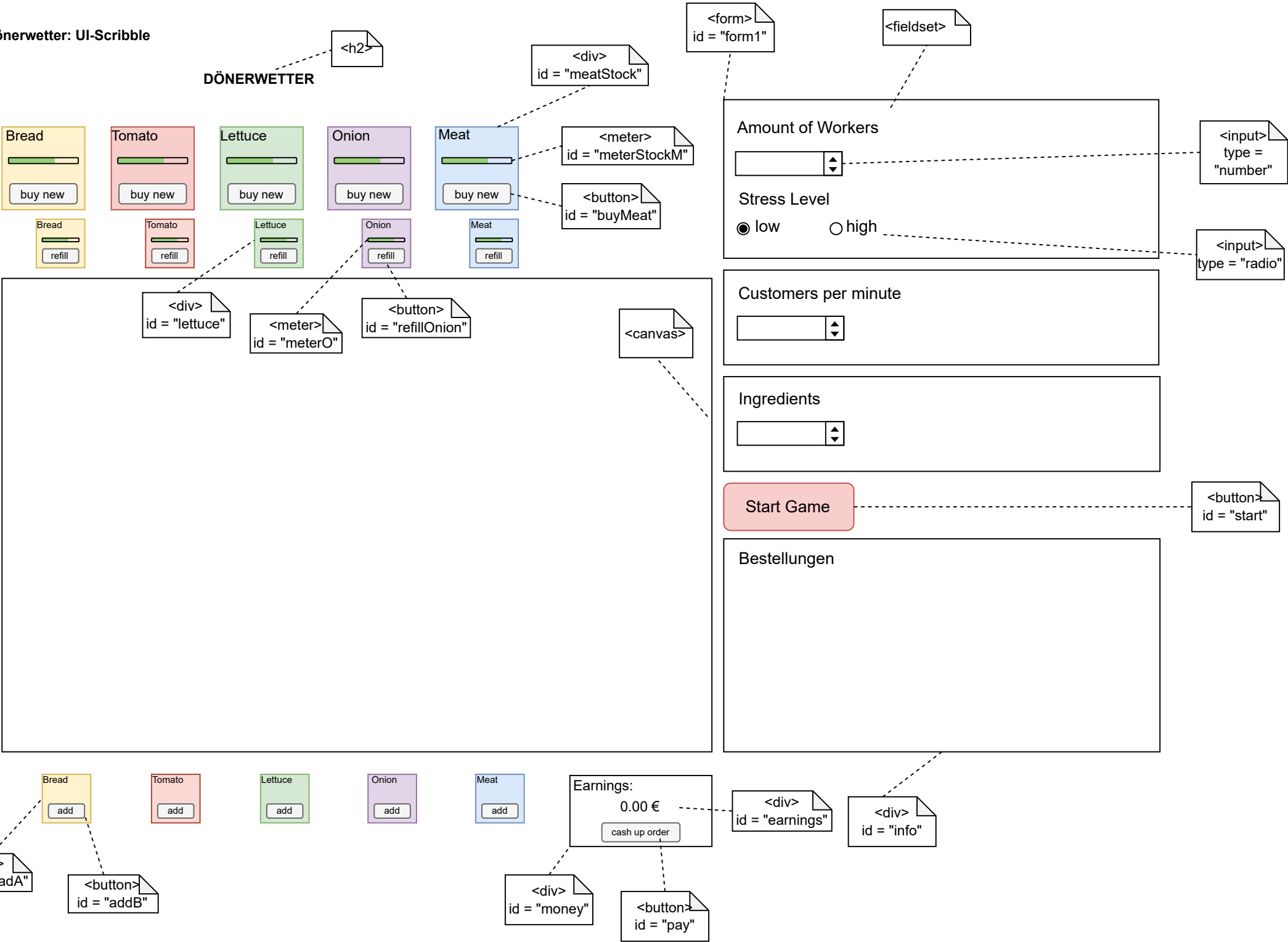
Endabgabe in der Veranstaltung  
Entwicklung Interaktiver Anwendungen ||

bei Prof. Jirka Dell'Oro-Friedl

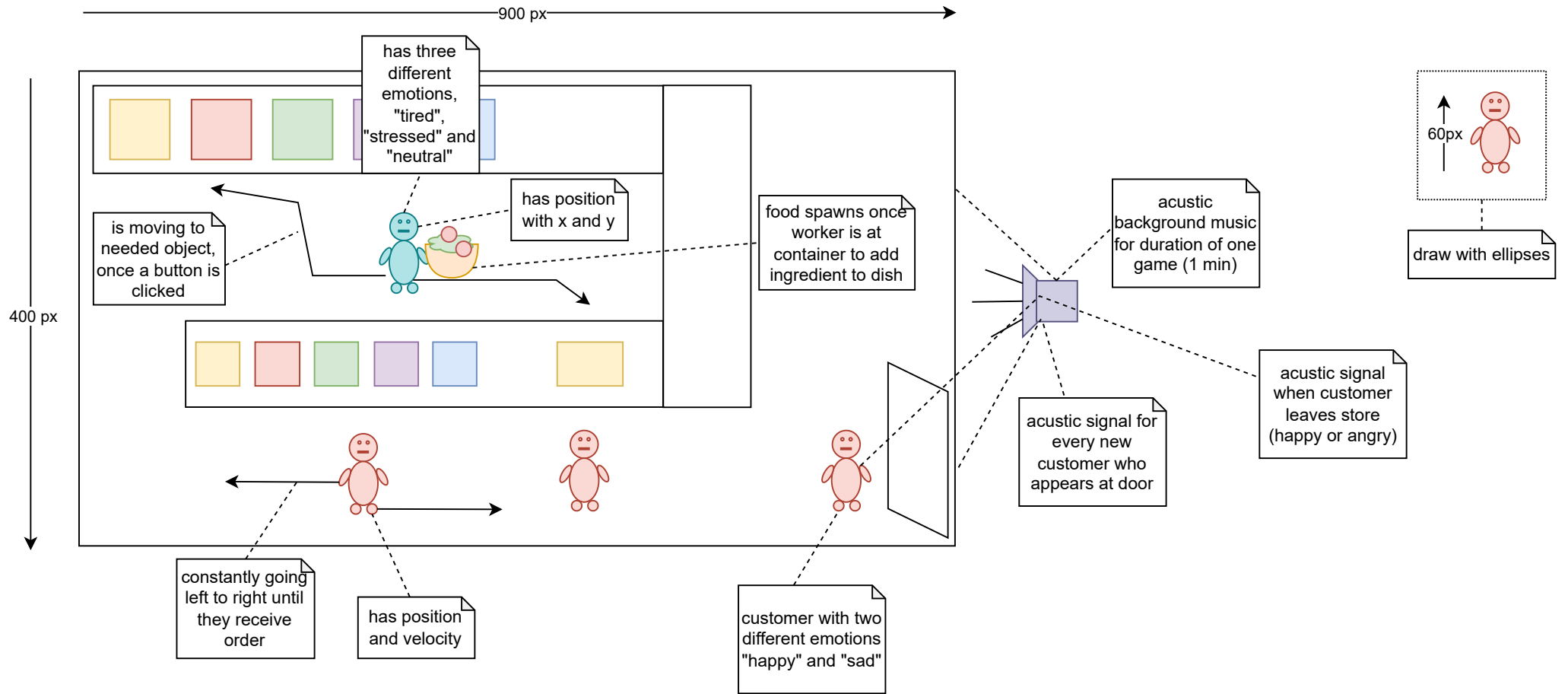
Fiona Virnich, MKB 4  
Matrikelnummer: 265115  
Vorgelegt am 15. Februar 2022  
Wintersemester 2021/2022  
In Zusammenarbeit mit Jasmin Basler

## Dönerwetter: Use-Case-Diagram

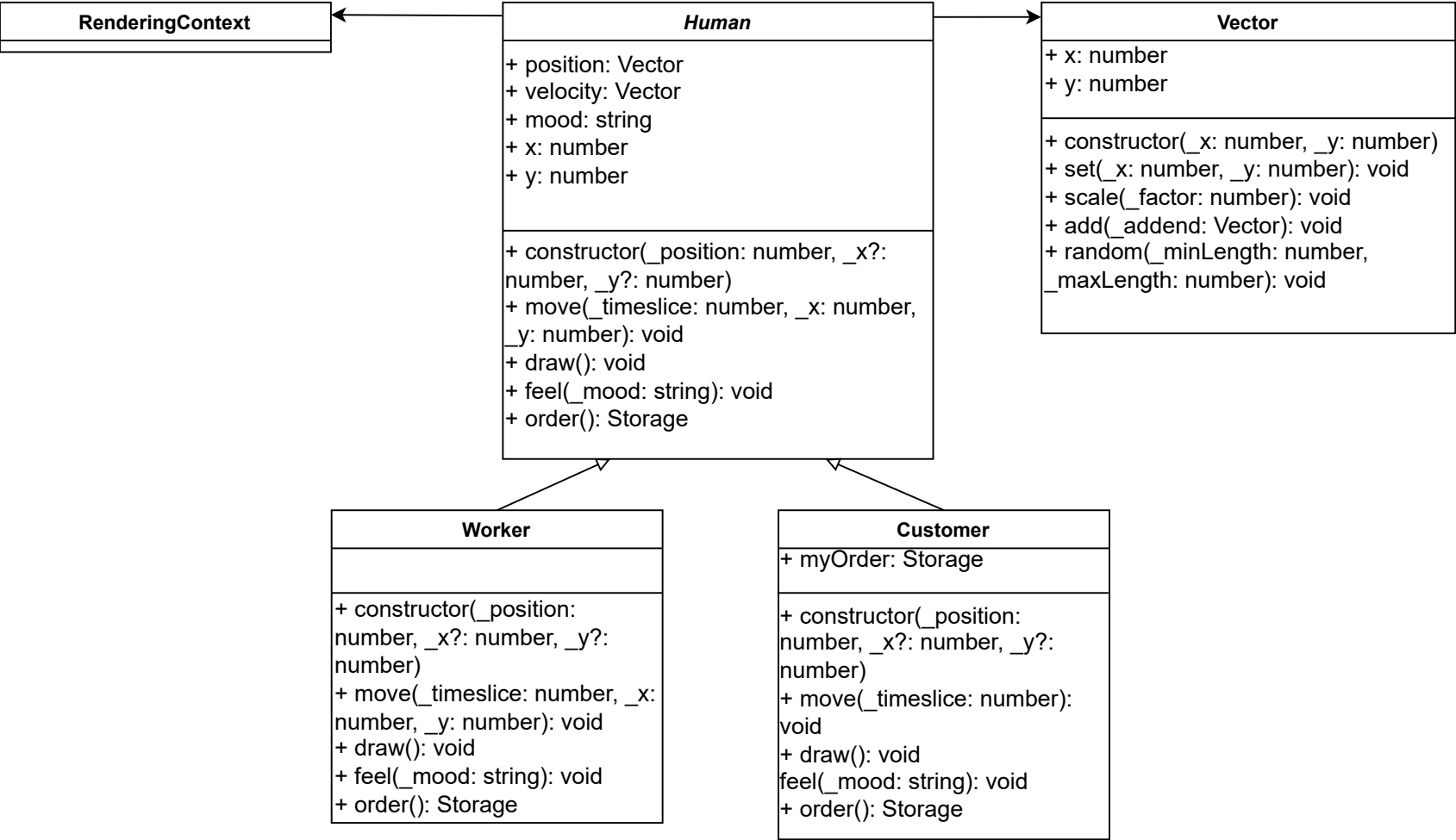


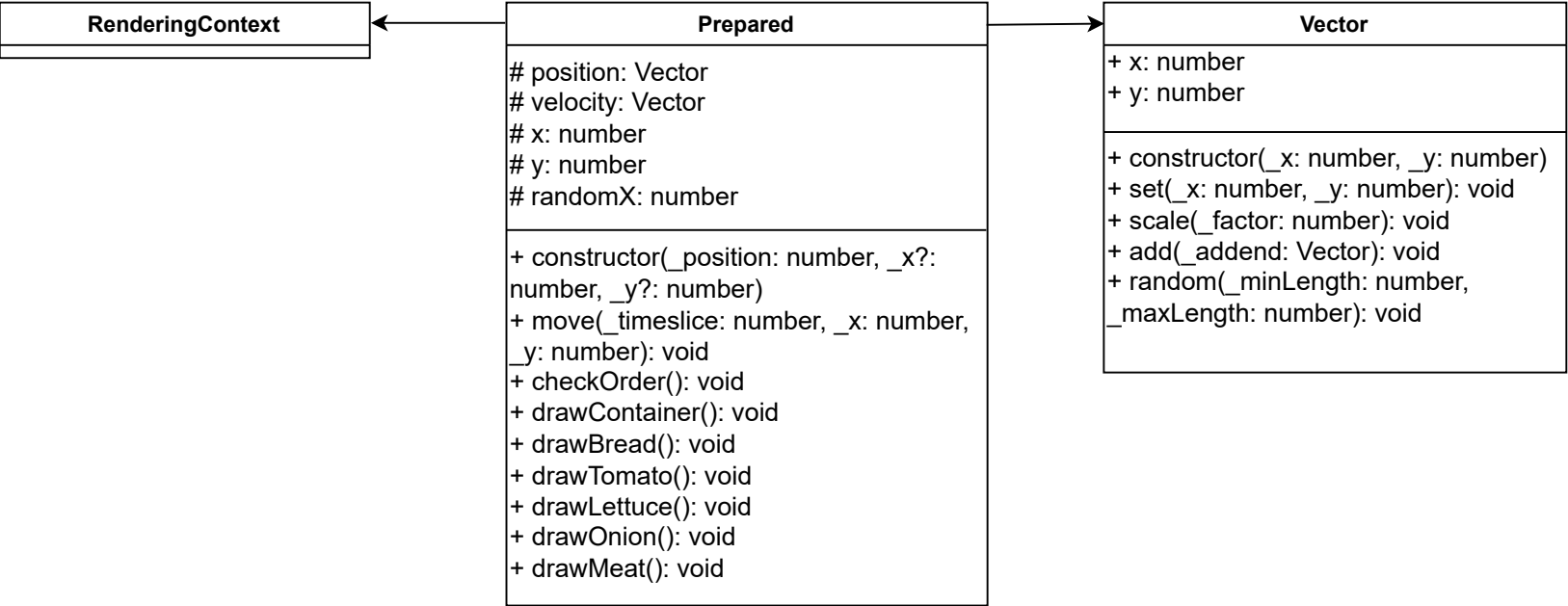


## Dönerwetter: UI-Scribble Canvas

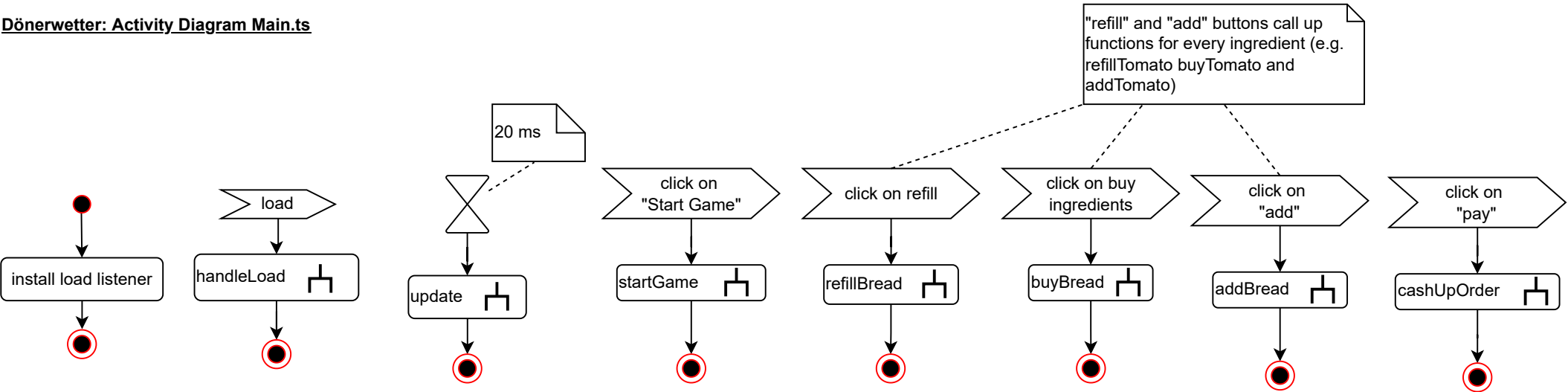


Dönerwetter: Class Diagram





Dönerwetter: Activity Diagram Main.ts



**Globals:**

crc2: CanvasRenderingContext2D  
let info: HTMLElement  
imageData: ImageData

workers: Worker[]  
customers: Customer[]  
orders: Storage[]  
ordersMade: Storage[]  
displayOrders : string[]  
drawOrders: Prepared[]  
currentCustomerAmount: number = 0  
earnings: number = 0  
happyScore: number = 0  
orderCorrect: boolean  
moodWorker: string  
moodCustomer: string

xOfWork: number  
yOfWork: number  
xOfWork2: number  
yOfWork2: number  
xOfCustomer: number  
yOfCustomer: number

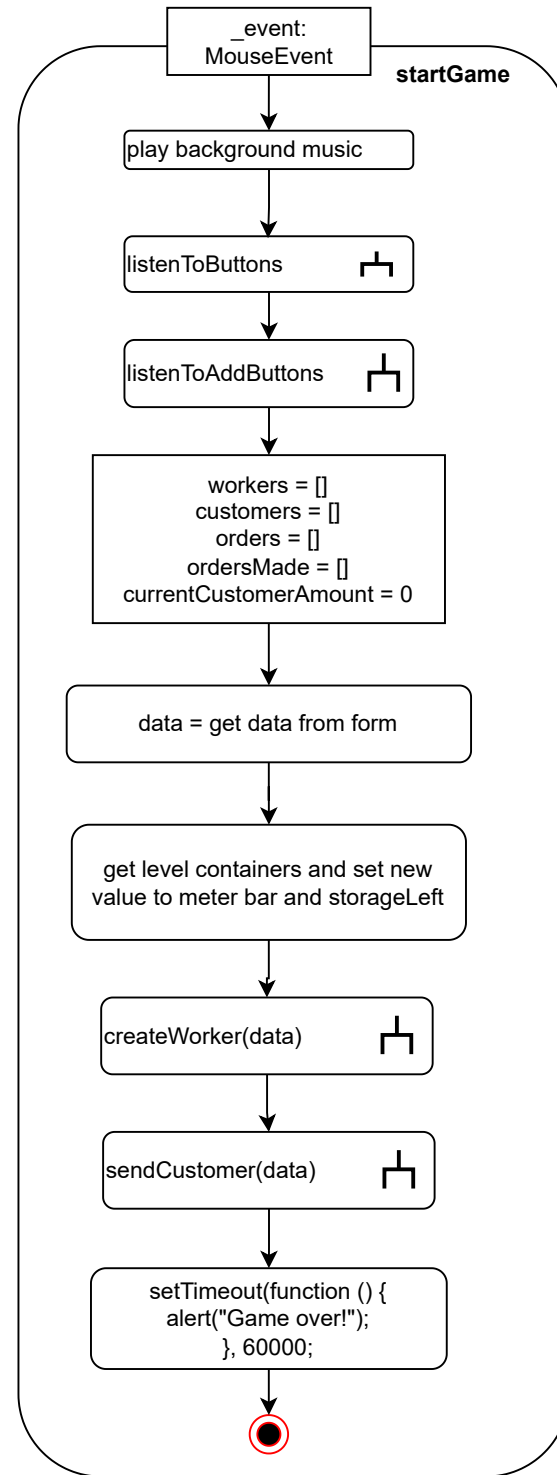
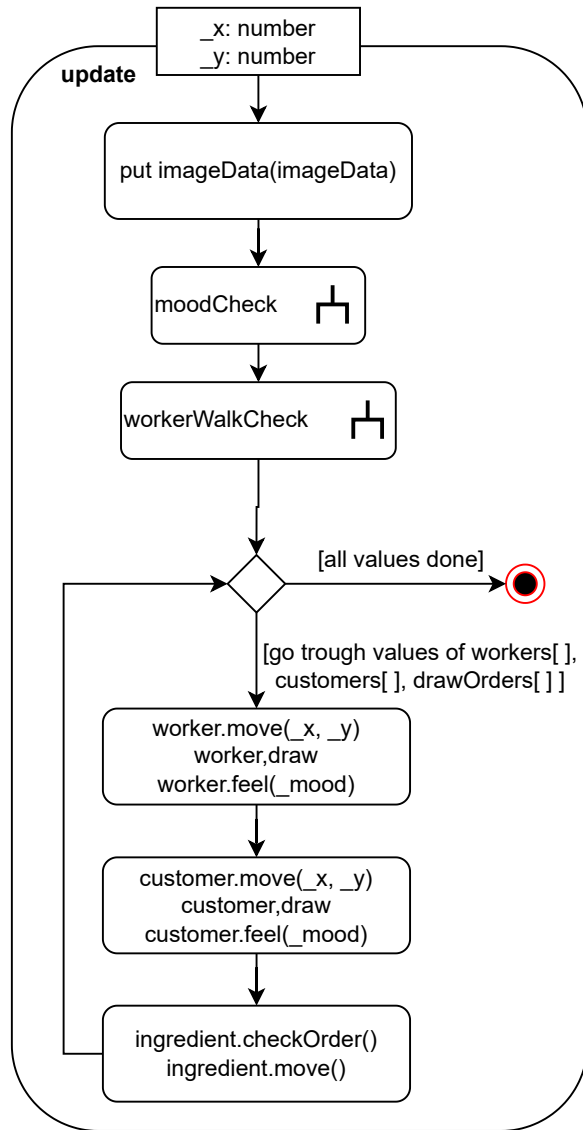
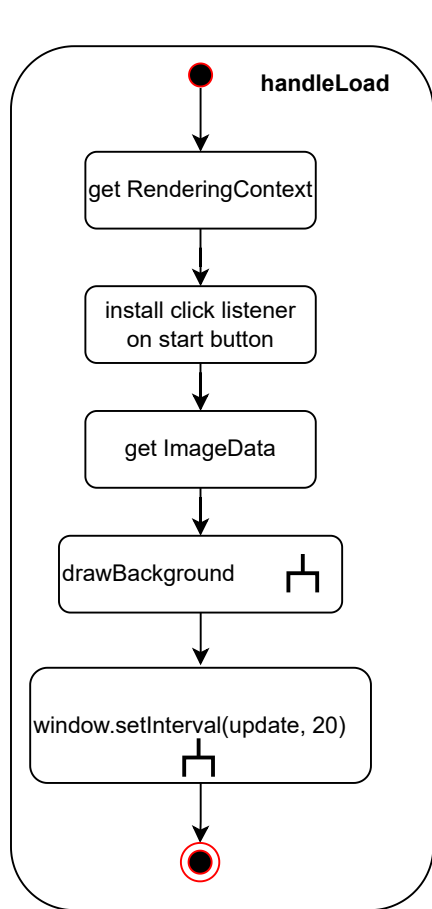
Storage
bread: number tomato: number lettuce: number onion: number meat: number

storageLeft
bread: 1000 tomato: 1000 lettuce: 1000 onion: 1000 meat: 1000

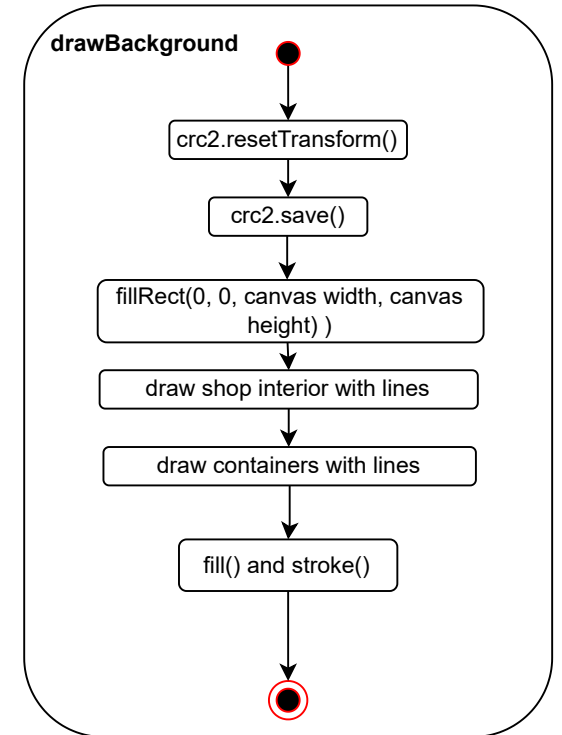
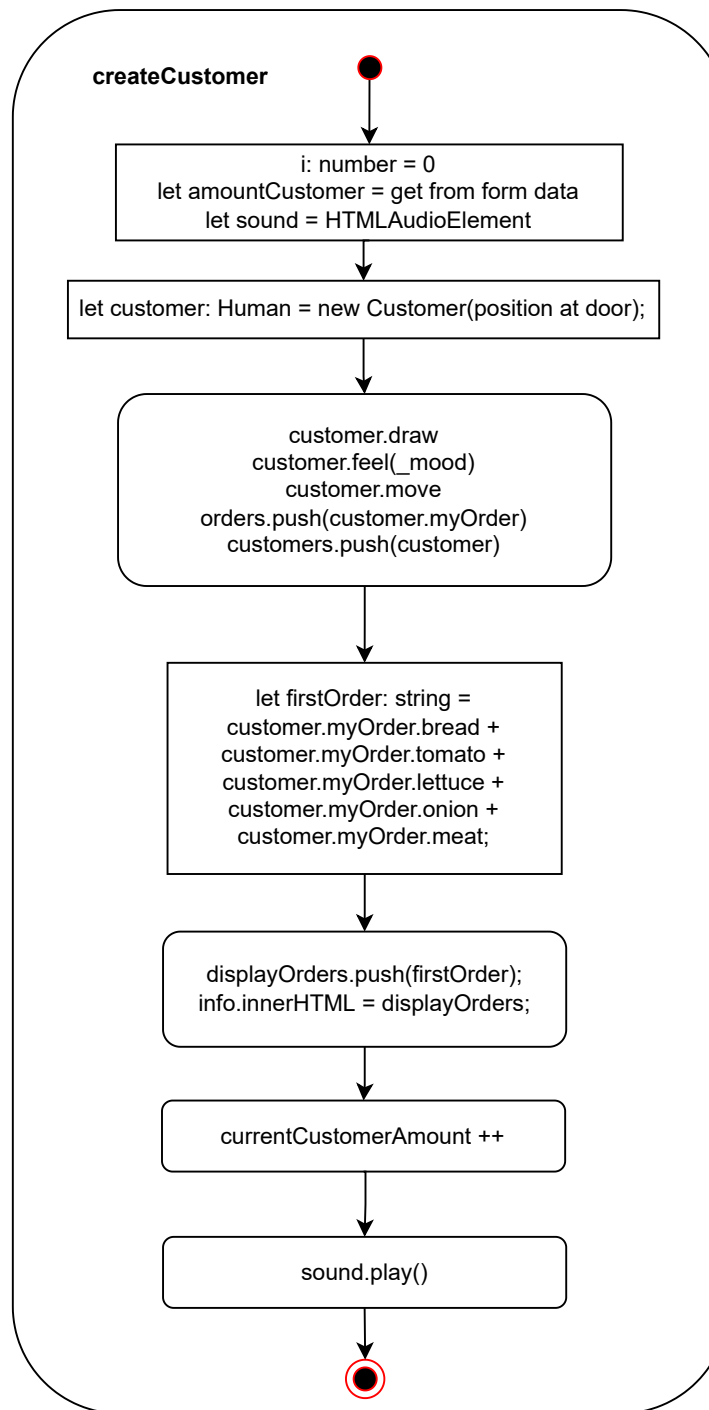
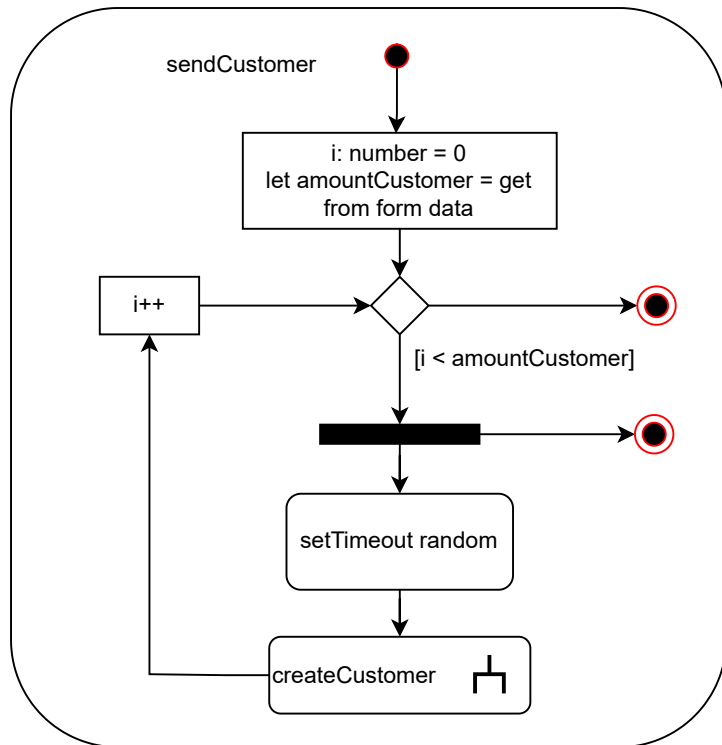
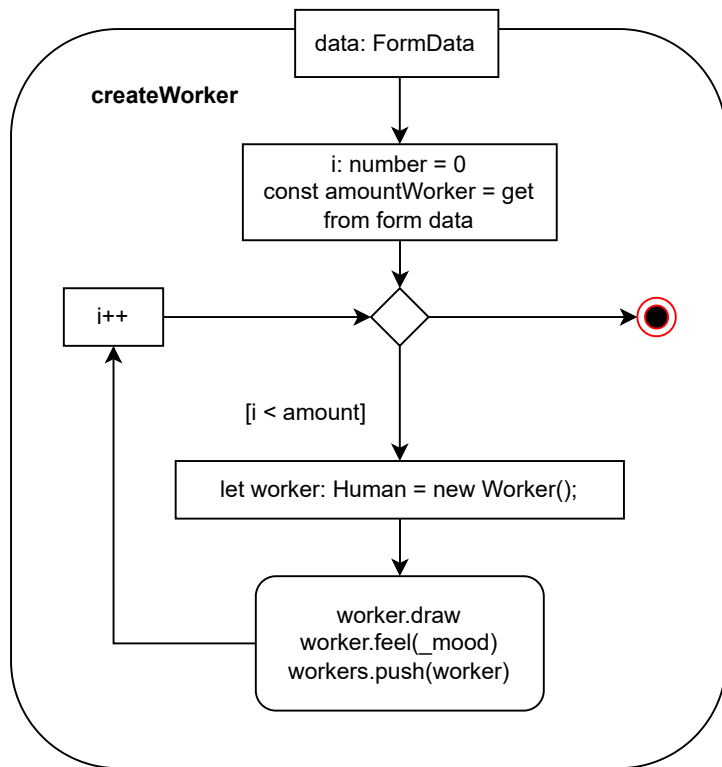
counterLeft
bread: 80 tomato: 80 lettuce: 80 onion: 80 meat: 80

currentOrder
bread: 0 tomato: 0 lettuce: 0 onion: 0 meat: 0

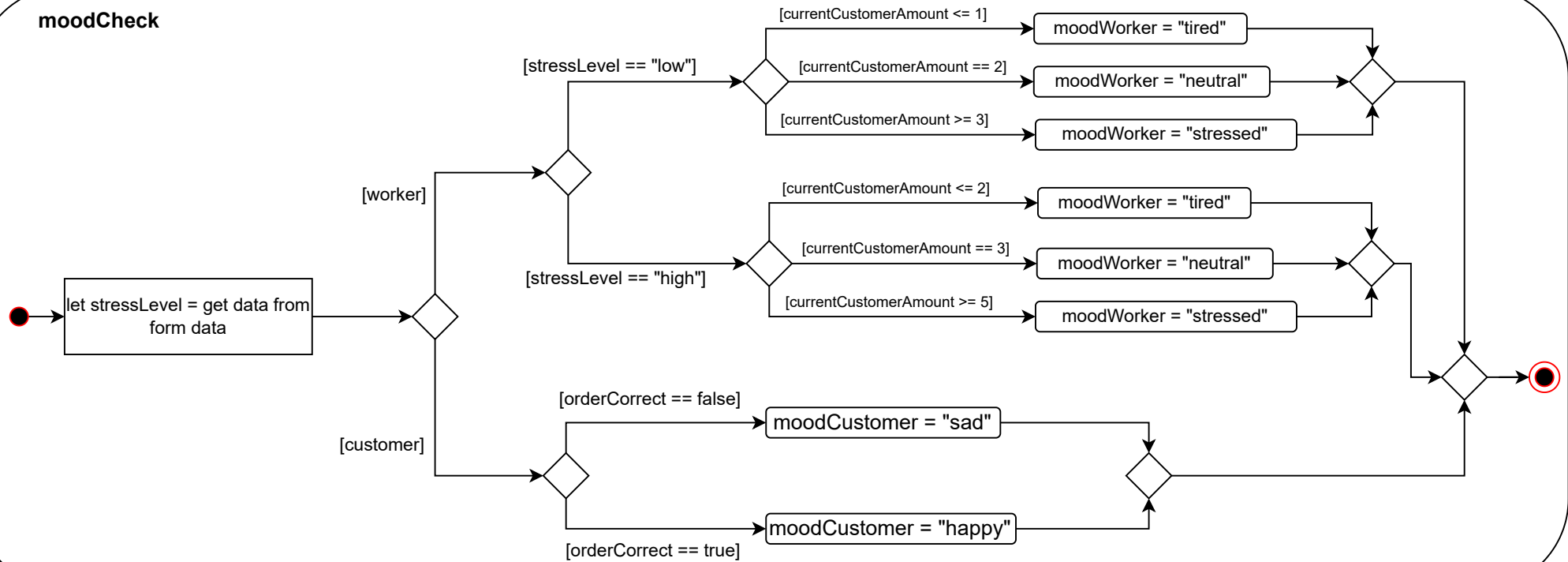
for more variety of dishes create new functions for new ingredient, create a new method in class Prepared and add ingredient as key in Storage --> use example of bread



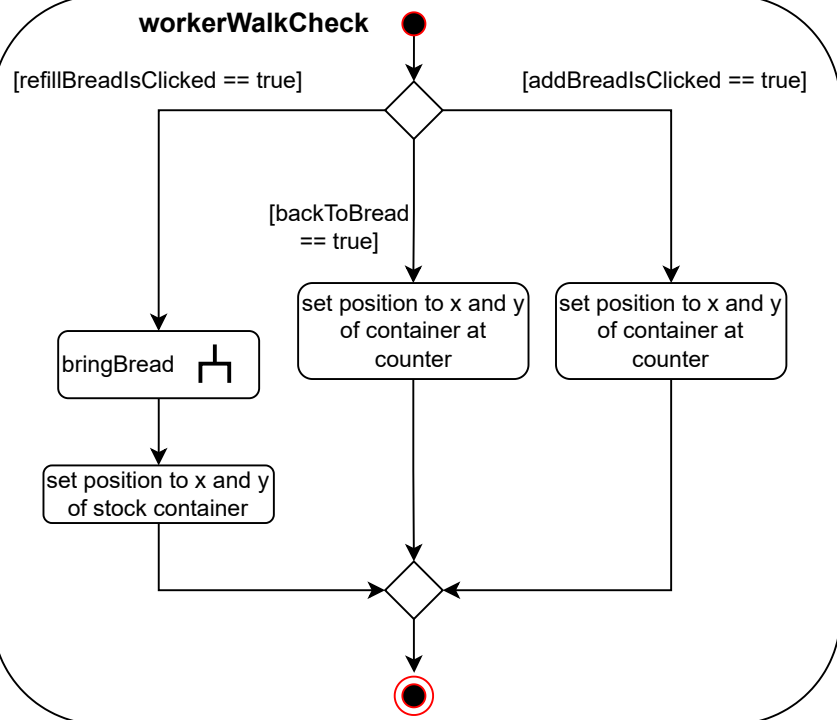




## moodCheck



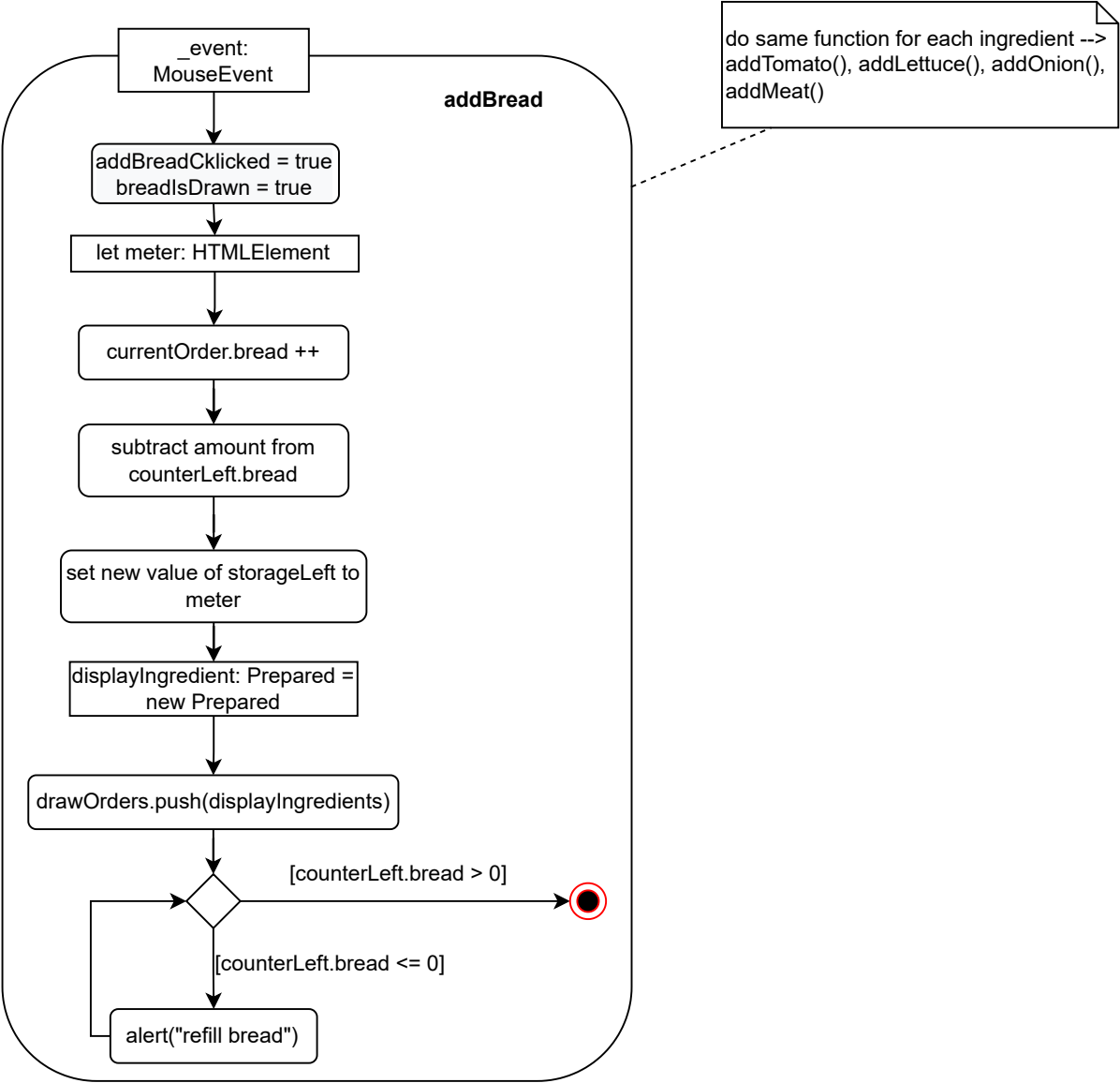
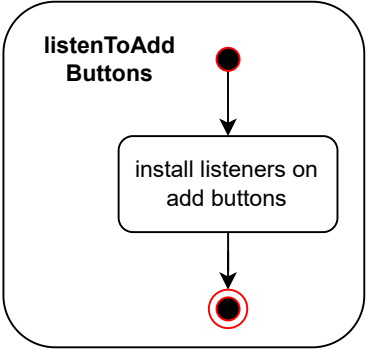
## workerWalkCheck



**Dönerwetter: Activity Diagram addIngredients.ts**

```
let addBreadIsClicked: boolean = false
let addTomatolsClicked: boolean = false
let addLettucelsClicked: boolean = false
let addOnionIsClicked: boolean = false
let addMeatIsClicked: boolean = false

let BreadIsDrawn: boolean = false
let TomatolsDrawn: boolean = false
let LettucelsDrawn: boolean = false
let OnionIsDrawn: boolean = false
let MeatIsDrawn: boolean = false
```



## Dönerwetter: Activity Diagram container.ts

```
let refillBreadIsClicked: boolean = false
let refillTomatolsClicked: boolean = false
let refillLettucelsClicked: boolean = false
let refillOnionIsClicked: boolean = false
let refillMeatIsClicked: boolean = false
```

```
let backToBread: boolean = false
let backToTomato: boolean = false
let backToLettuce: boolean = false
let backToOnion: boolean = false
let backToMeat: boolean = false
```

### listenToButtons

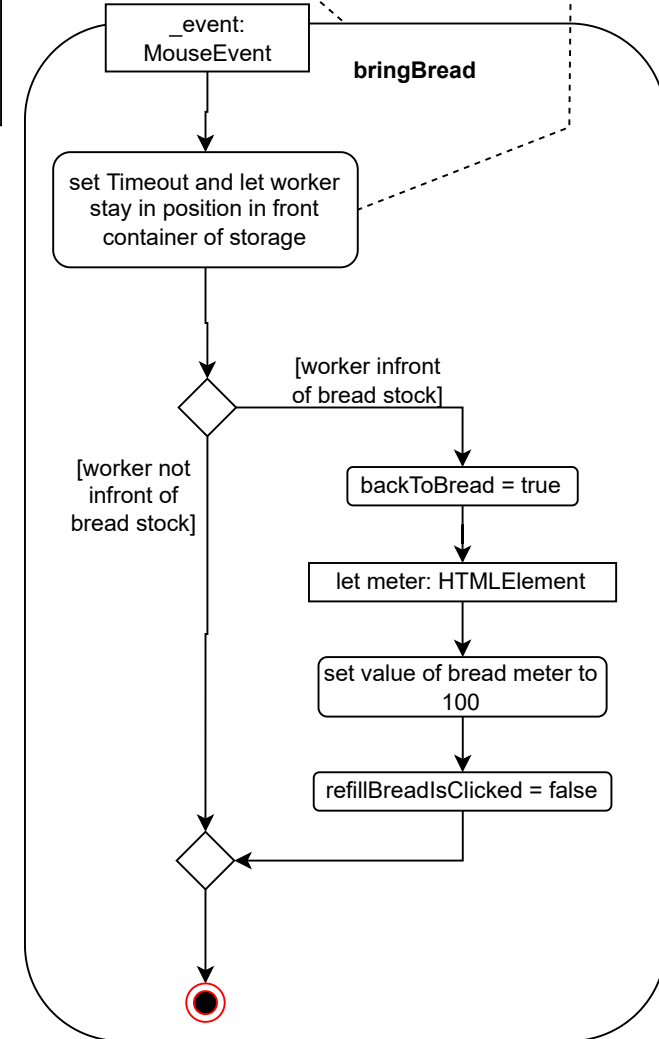
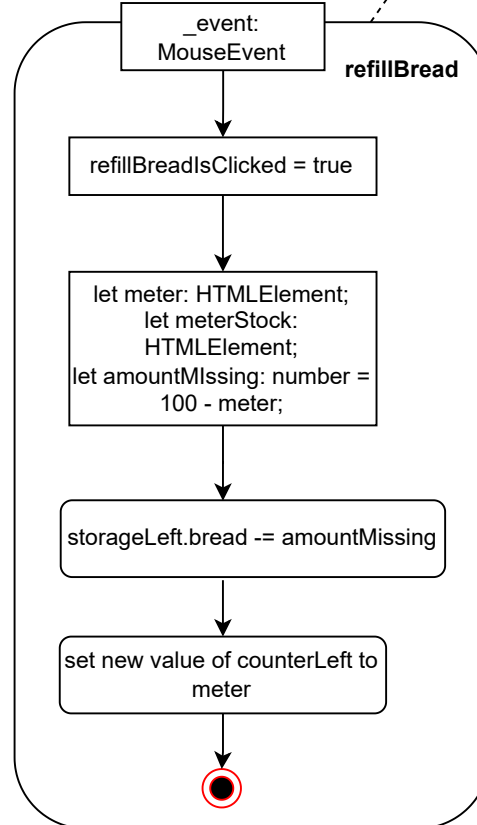
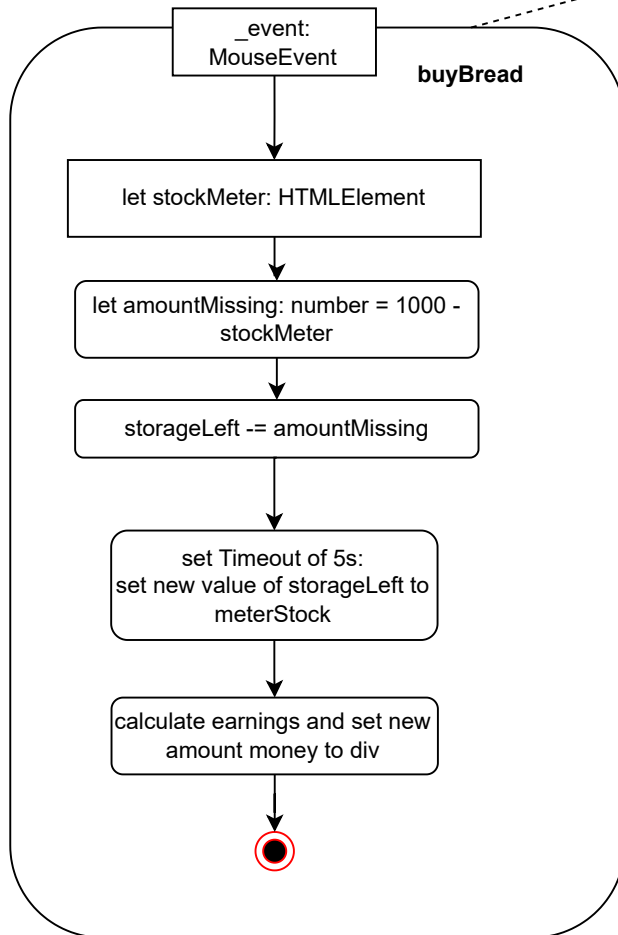
install listeners on  
refill and buy buttons

do same function for each ingredient -->  
buyTomato(), buyLettuce(), buyOnion(),  
buyMeat()

do same function for  
each ingredient -->  
refillTomato(),  
refillLettuce(),  
refillOnion(),  
refillMeat()

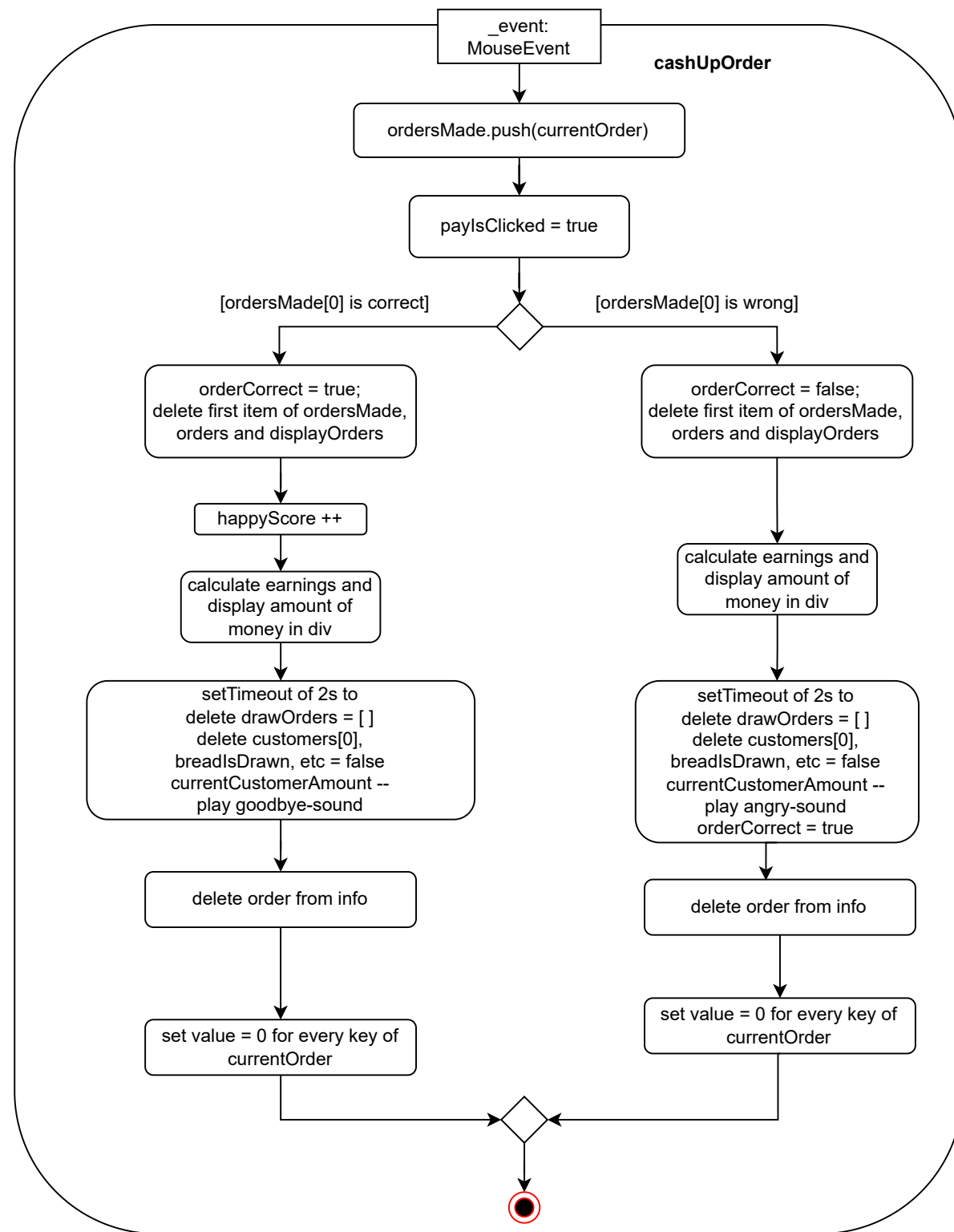
do same function for  
each ingredient -->  
bringTomato(),  
bringLettuce(),  
bringOnion(),  
bringMeat()

set different value of  
Timeout for each  
ingredient  
--> e.g. worker waits  
longer in front of onion  
container than in front  
of lettuce container

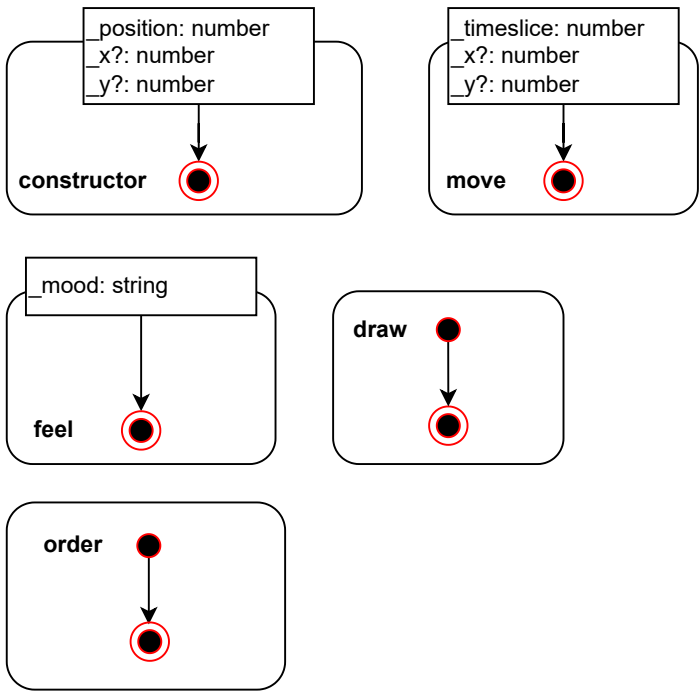


## Dönerwetter: Activity Diagram pay.ts

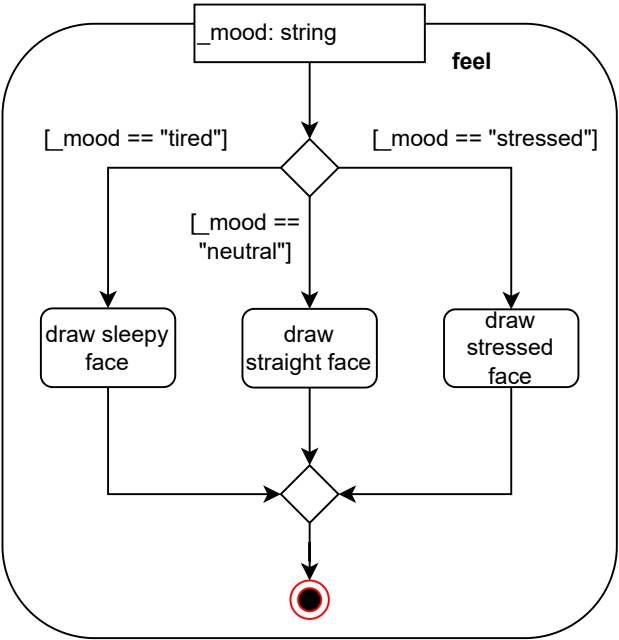
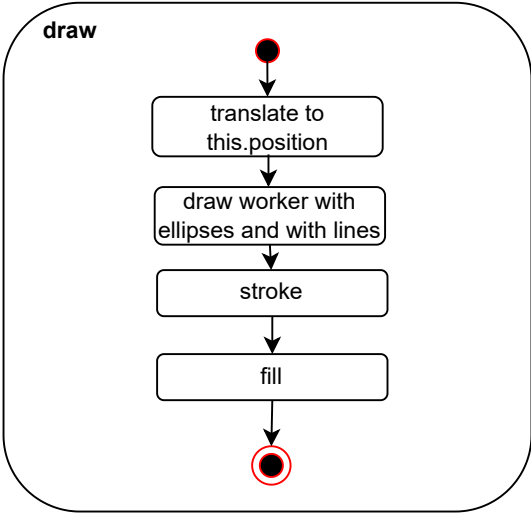
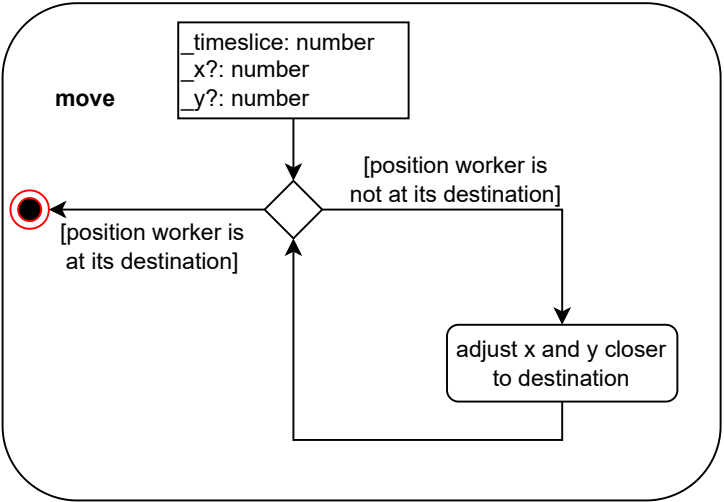
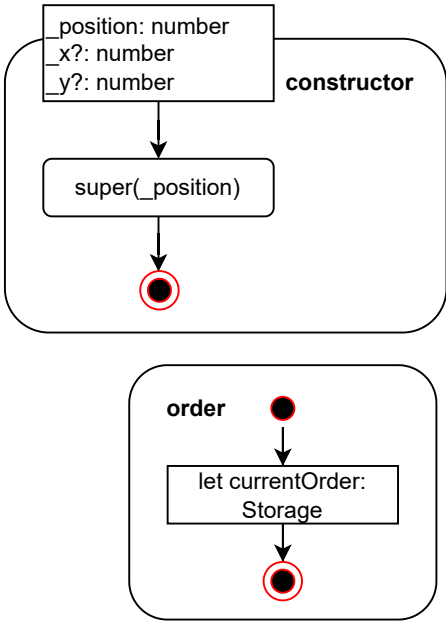
let payIsClicked : boolean = false



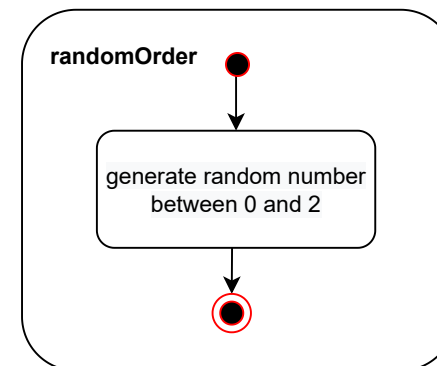
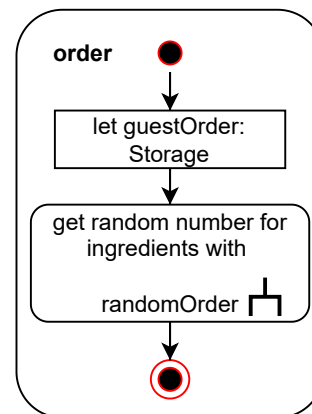
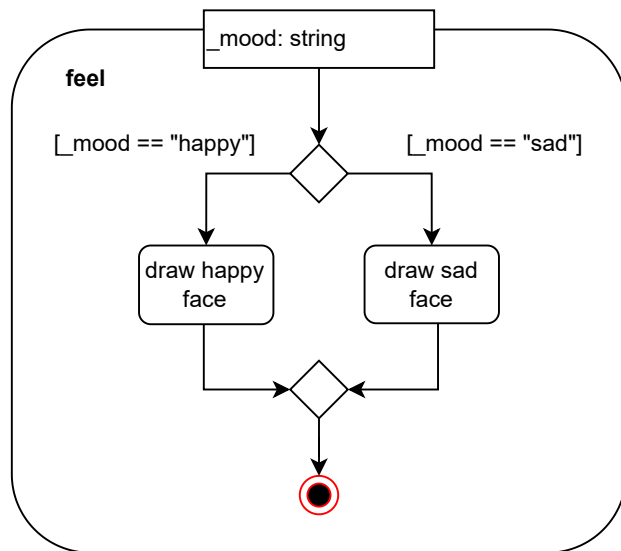
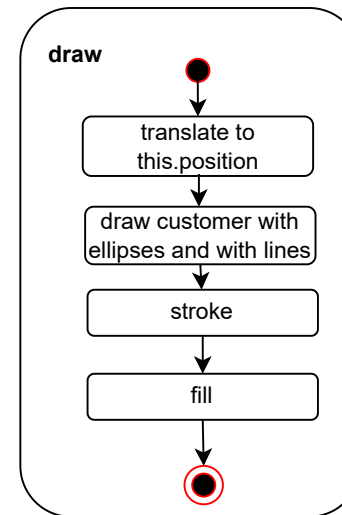
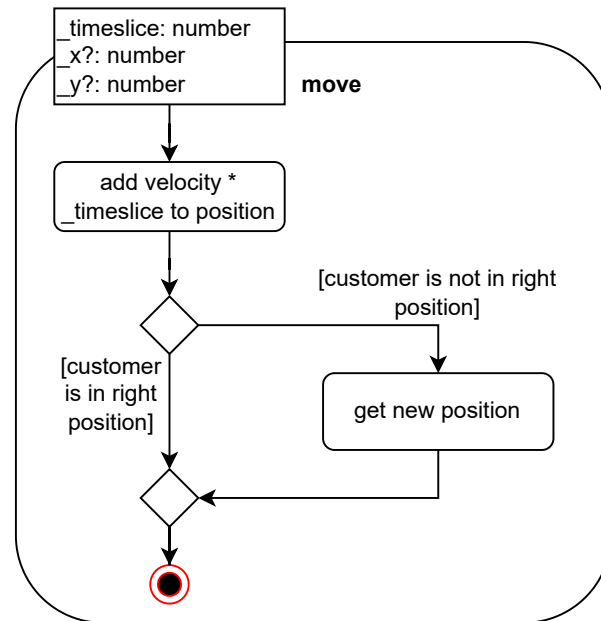
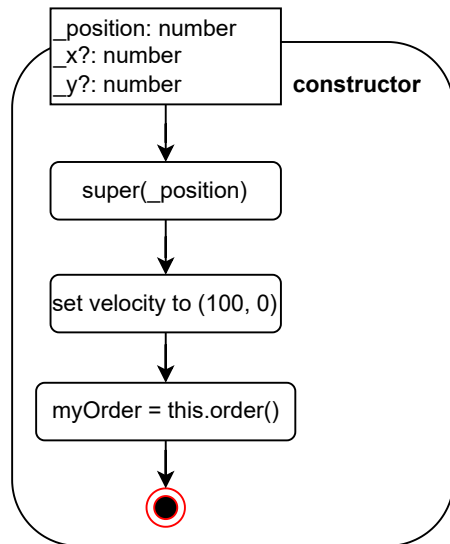
Dönerwetter: Activity Diagram - Human



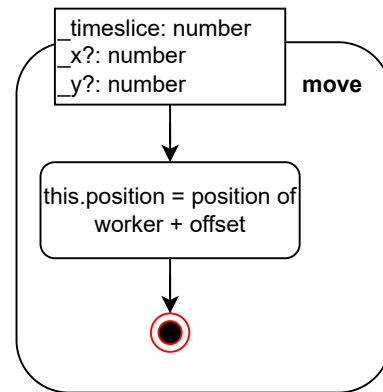
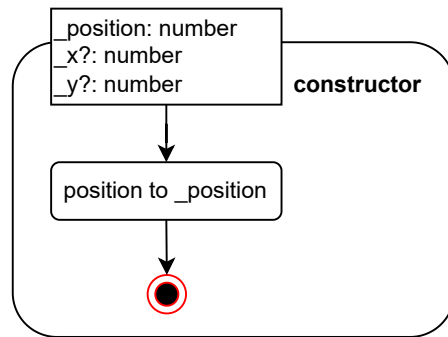
Dönerwetter: Activity Diagram - Worker



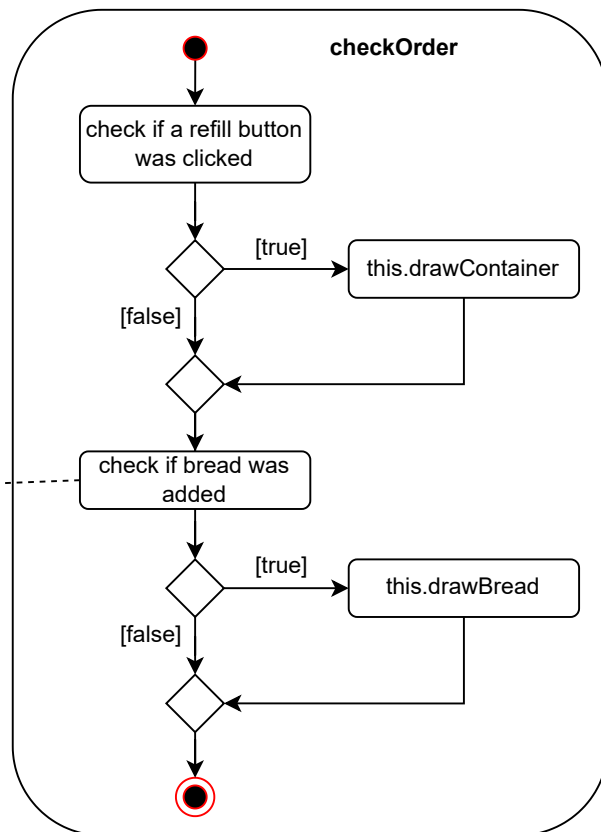
## Dönerwetter: Activity Diagram - Customer



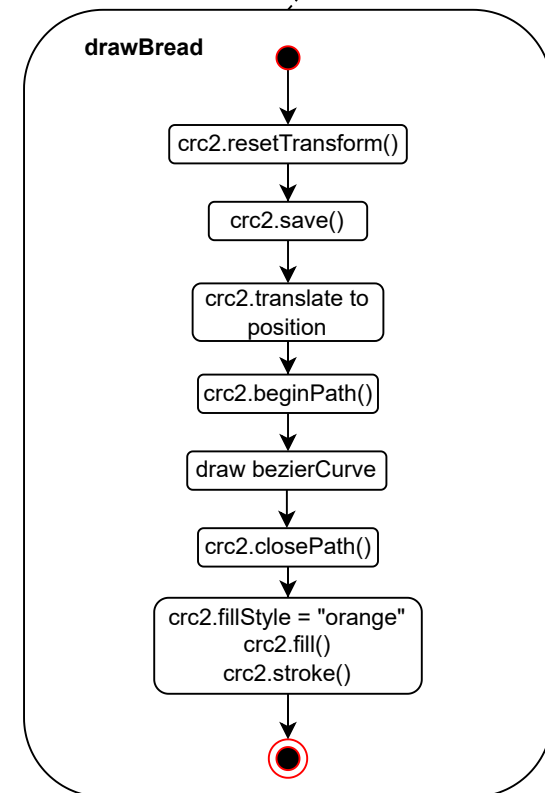
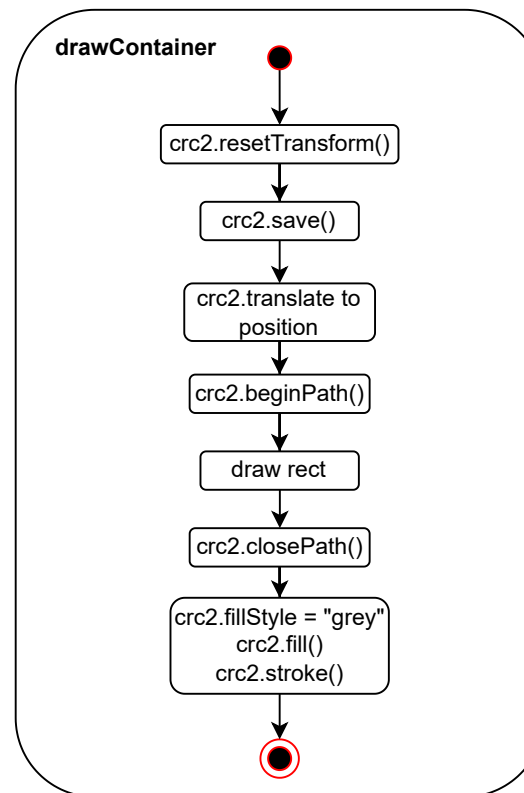
## Dönerwetter: Activity Diagram - Prepared



create same method for each ingredient --> drawTomato, drawLettuce, drawOnion and drawMeat work the same



check for each ingredient





## Dönerwetter: Activity Diagram - Vector

