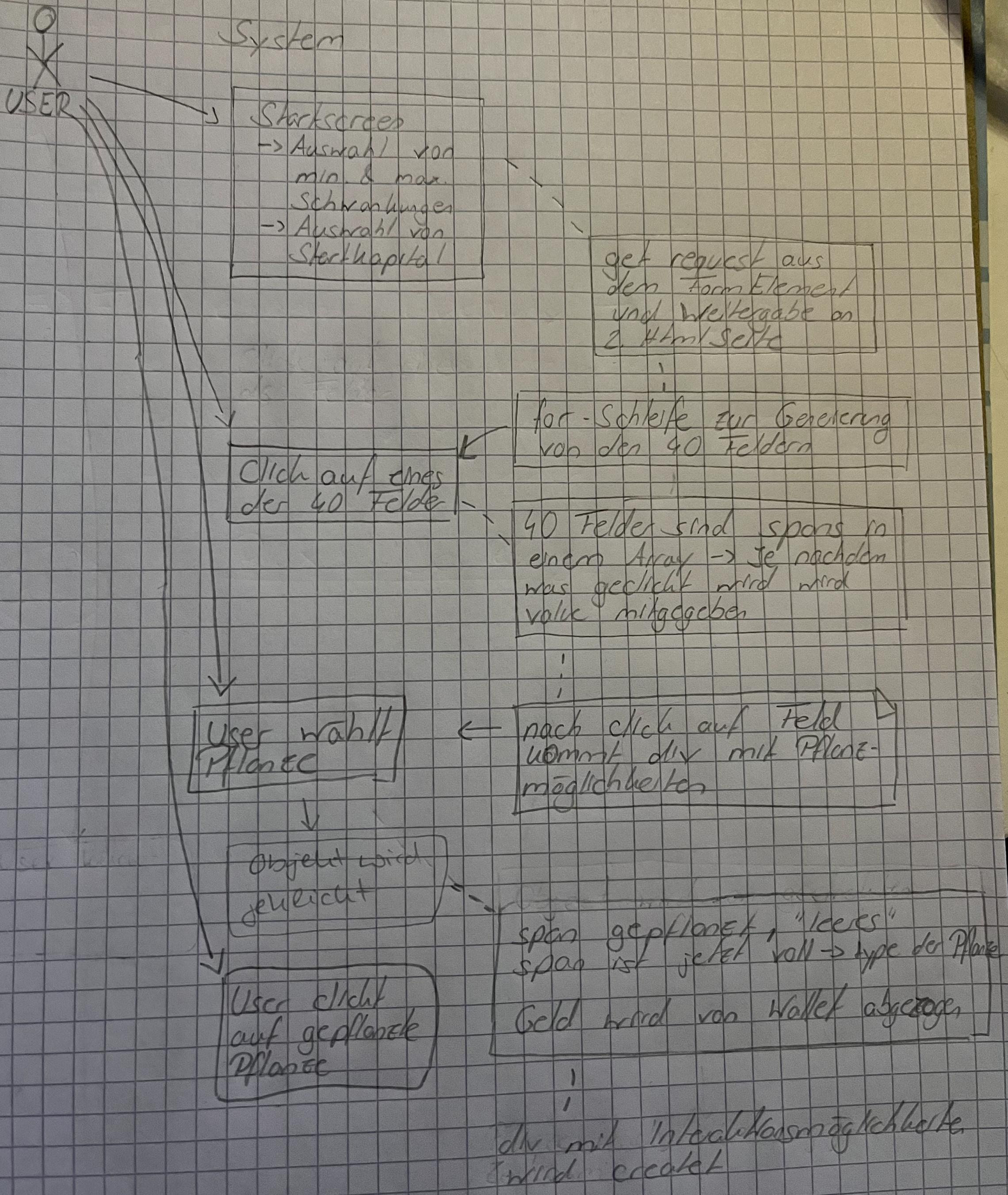


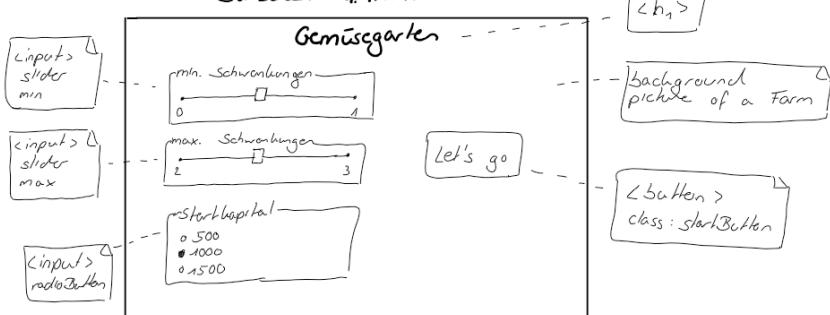
USE - CASE - Diagram



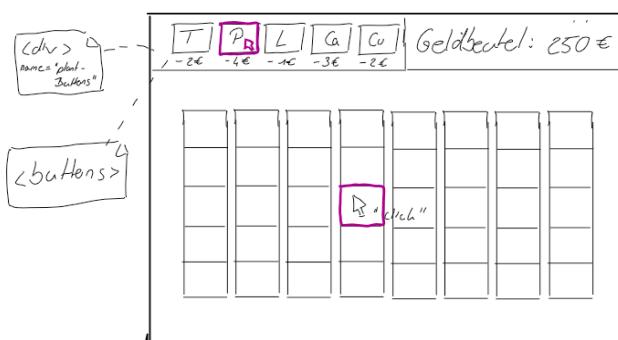
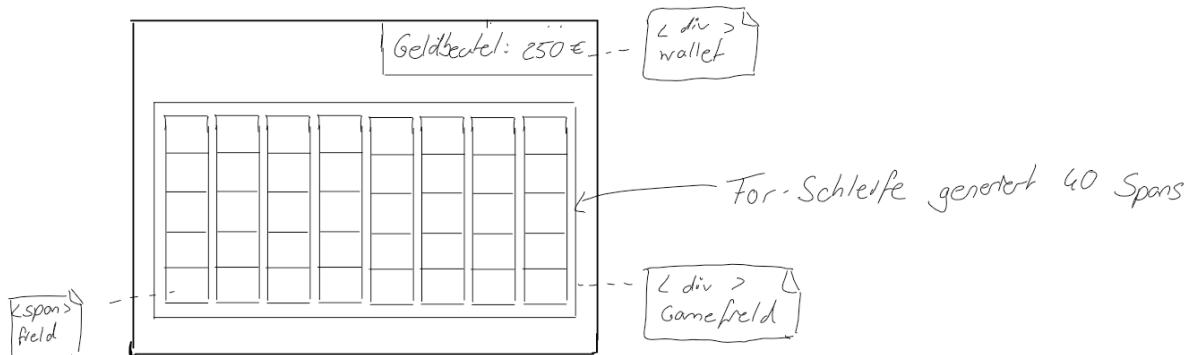
UI - Scribble

Mittwoch, 29. Juni 2022 16:05

Startscreen 1. HTML Seite



2. HTML Seite ---> übertragung mit "get request"

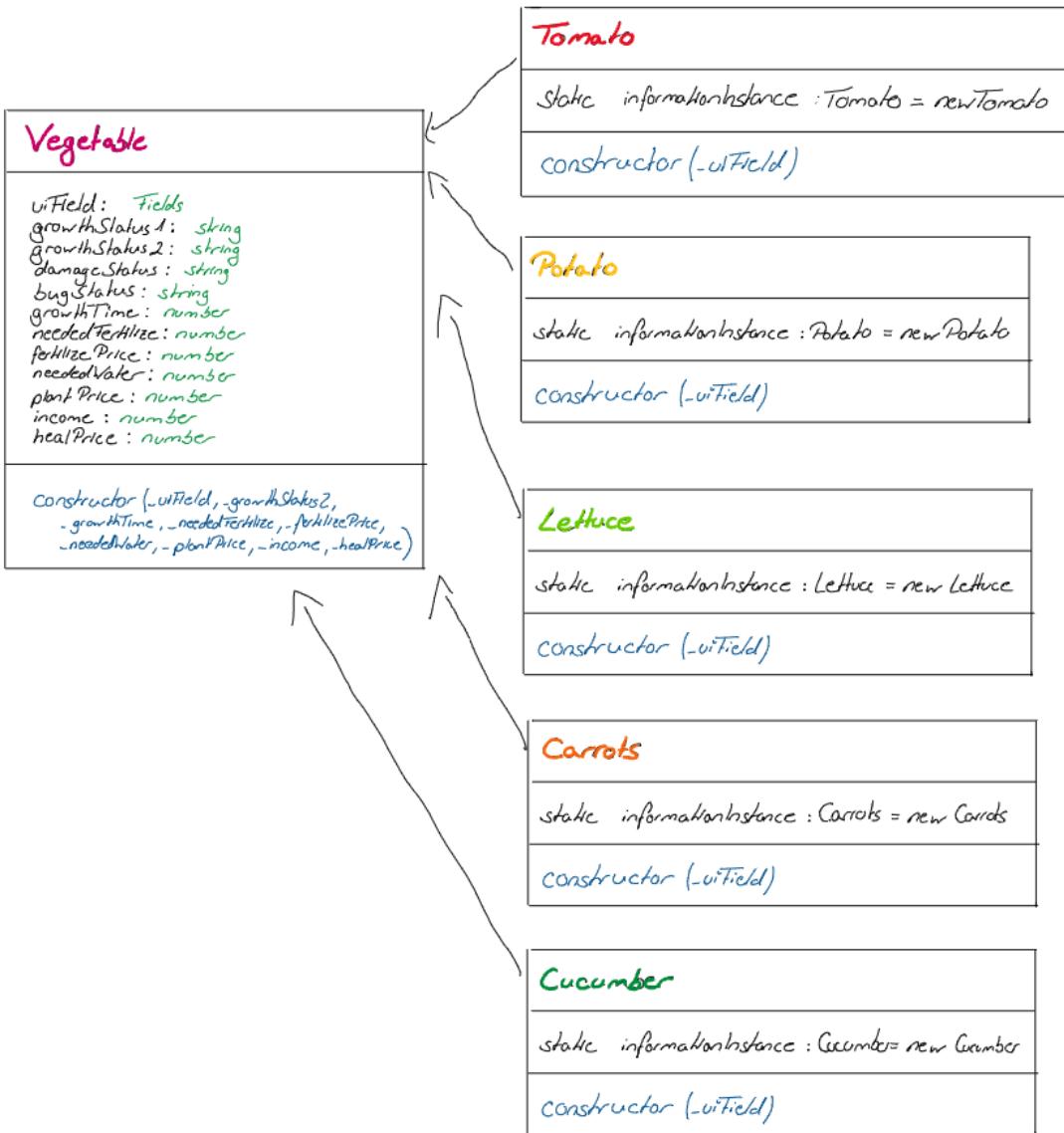


→ +5 €

→ shows information what the plant needs

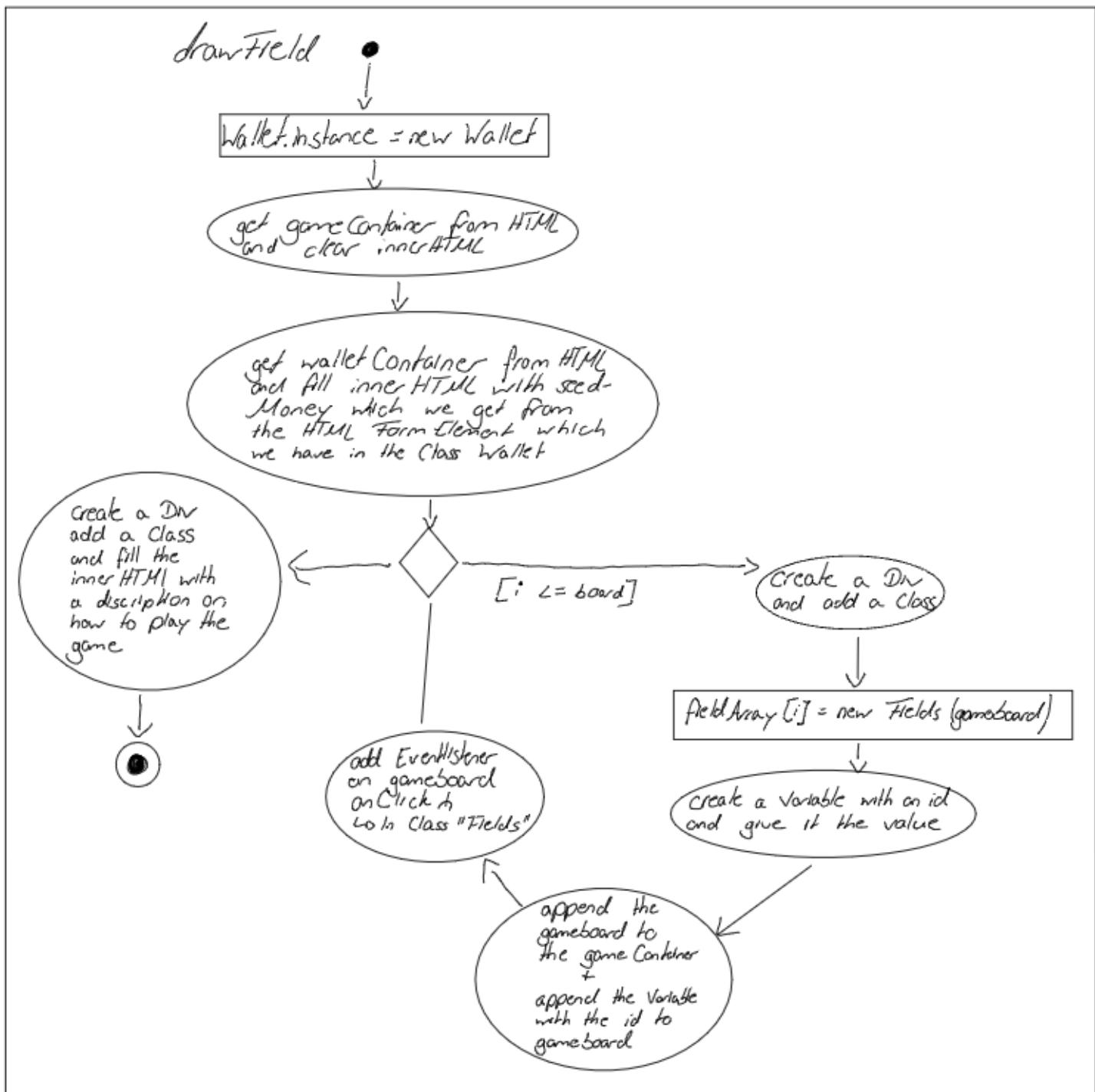
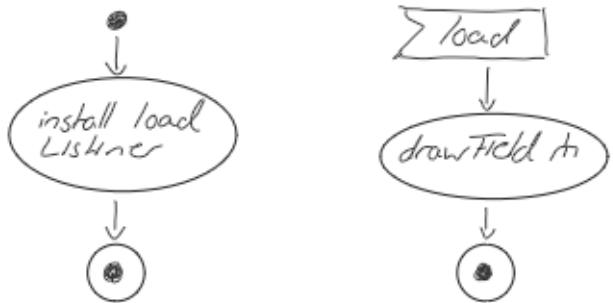
Fields
gameDiv : HTMLDivElement recentVegetable: Vegetable infoContainer: HTMLDivElement
Constructor (gameDivGiven)

Wallet
Static instance : Wallet urlParam : URLSearchParams minPrice: number maxPrice: number seedMoney: number inflationRatio: number
Constructor()



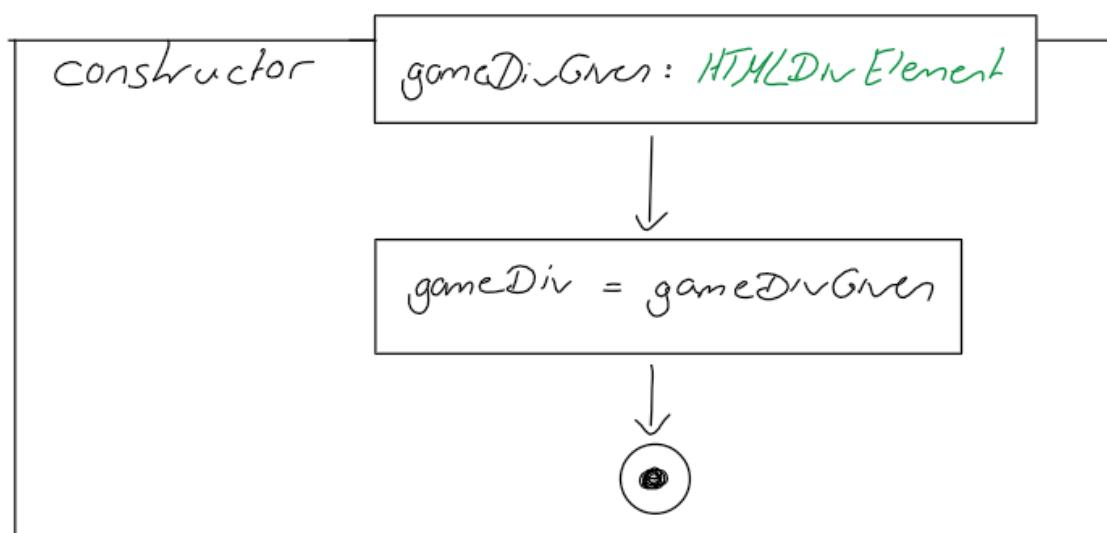
Activity Diagram - Main

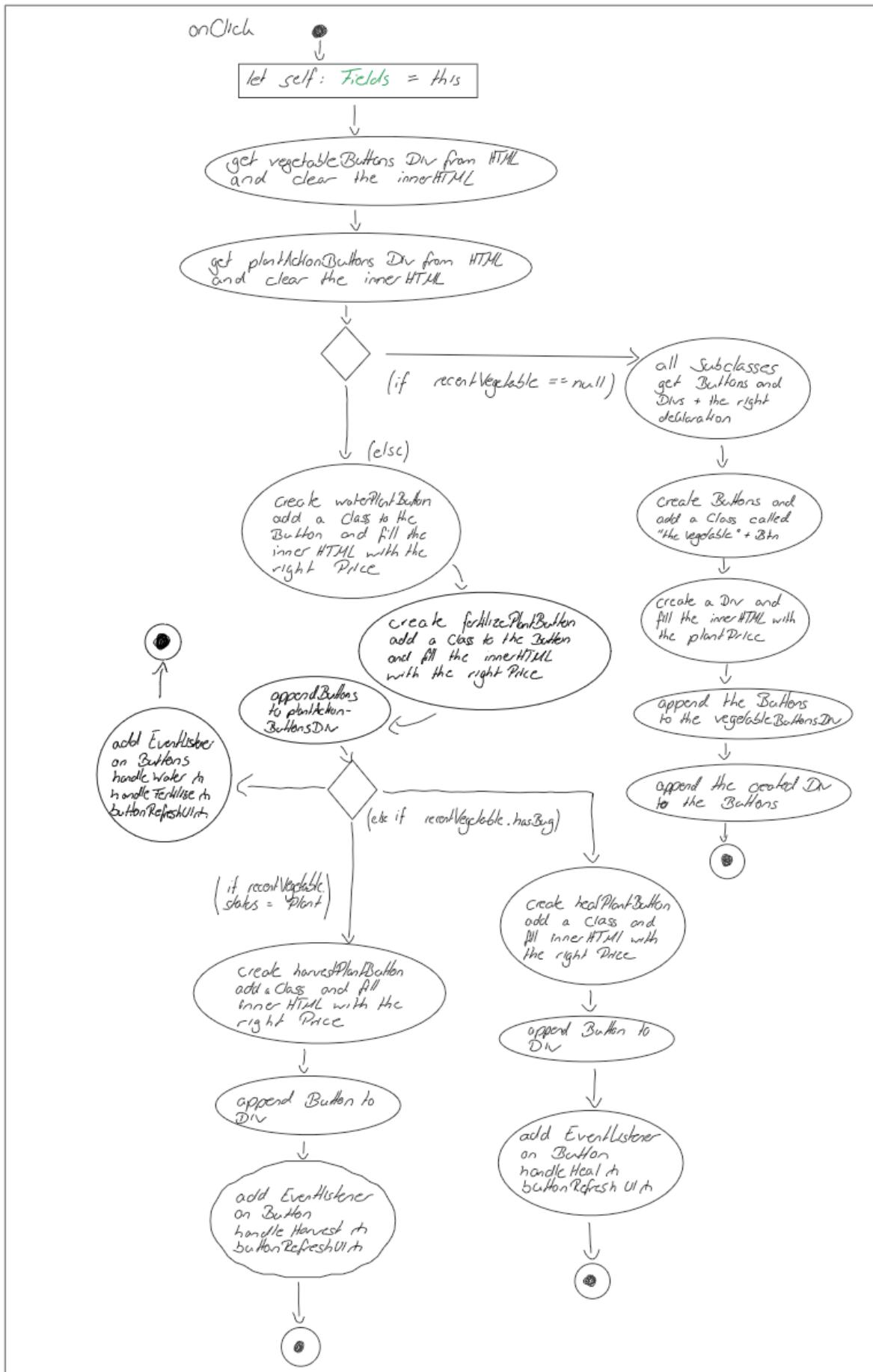
```
let board: number
let gameboard: HTMLDivElement
let gameContainer: HTMLDivElement
let fieldArray: Fields[] = []
let walletContainer: HTMLDivElement
let textContainer: HTMLDivElement
```

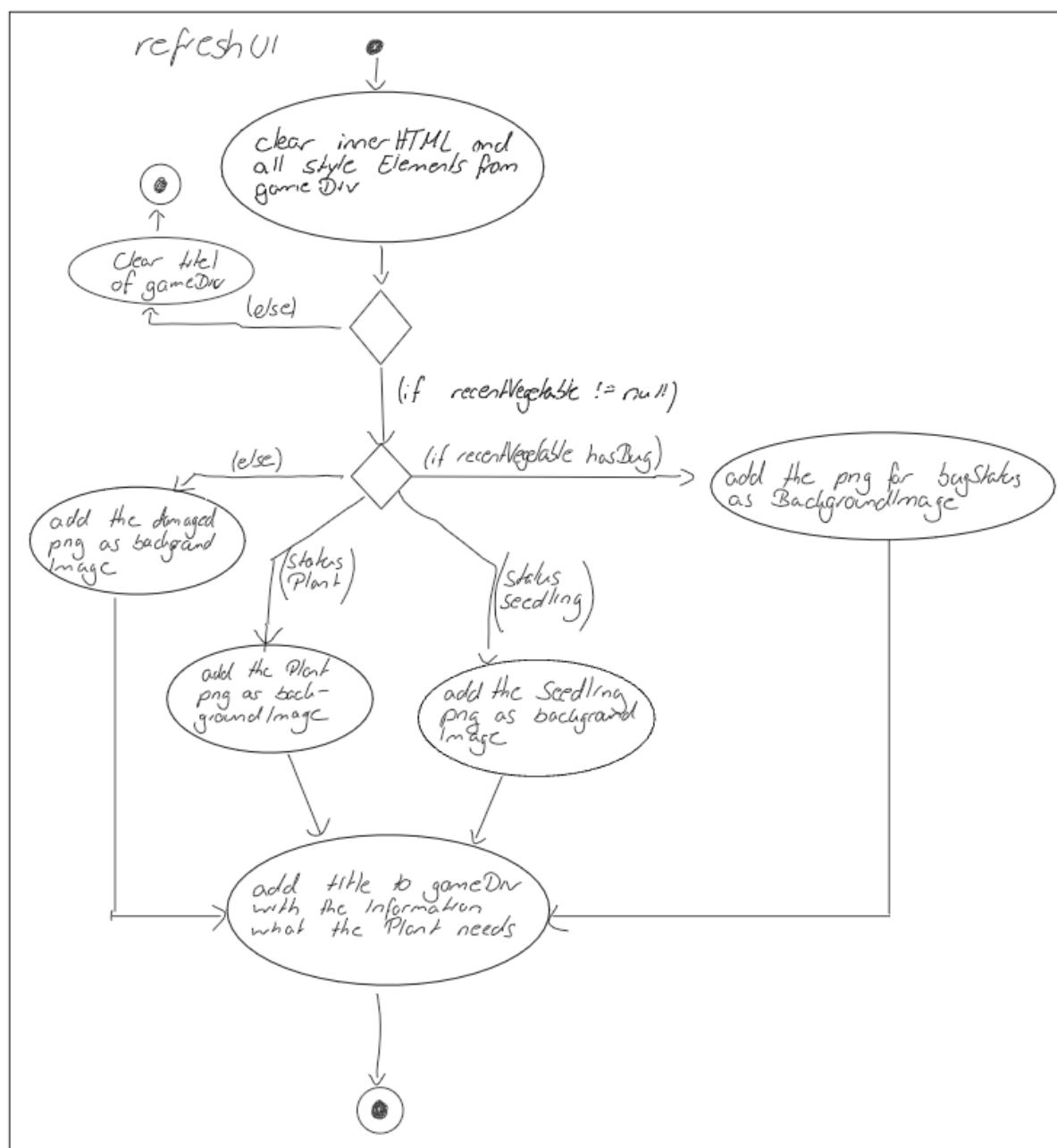
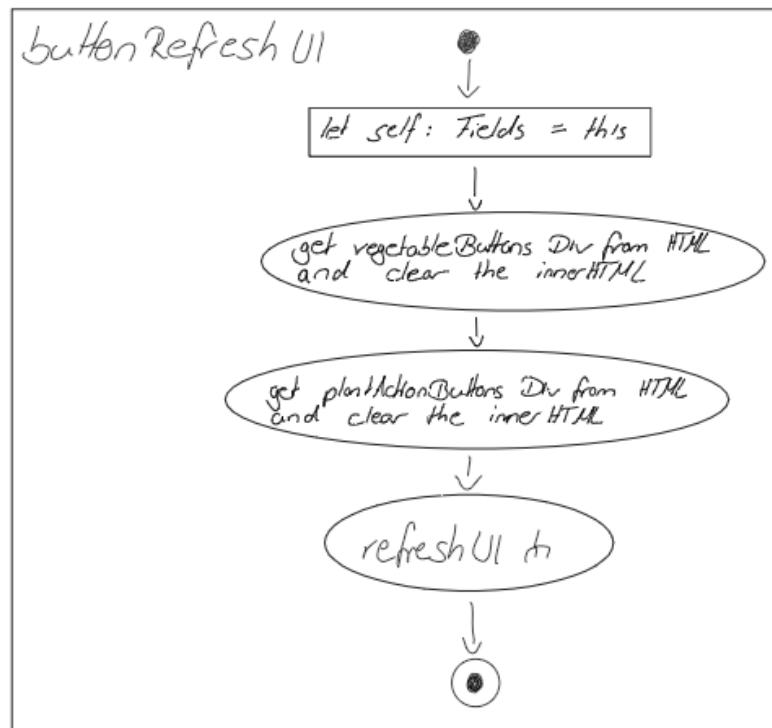


Class - Fields

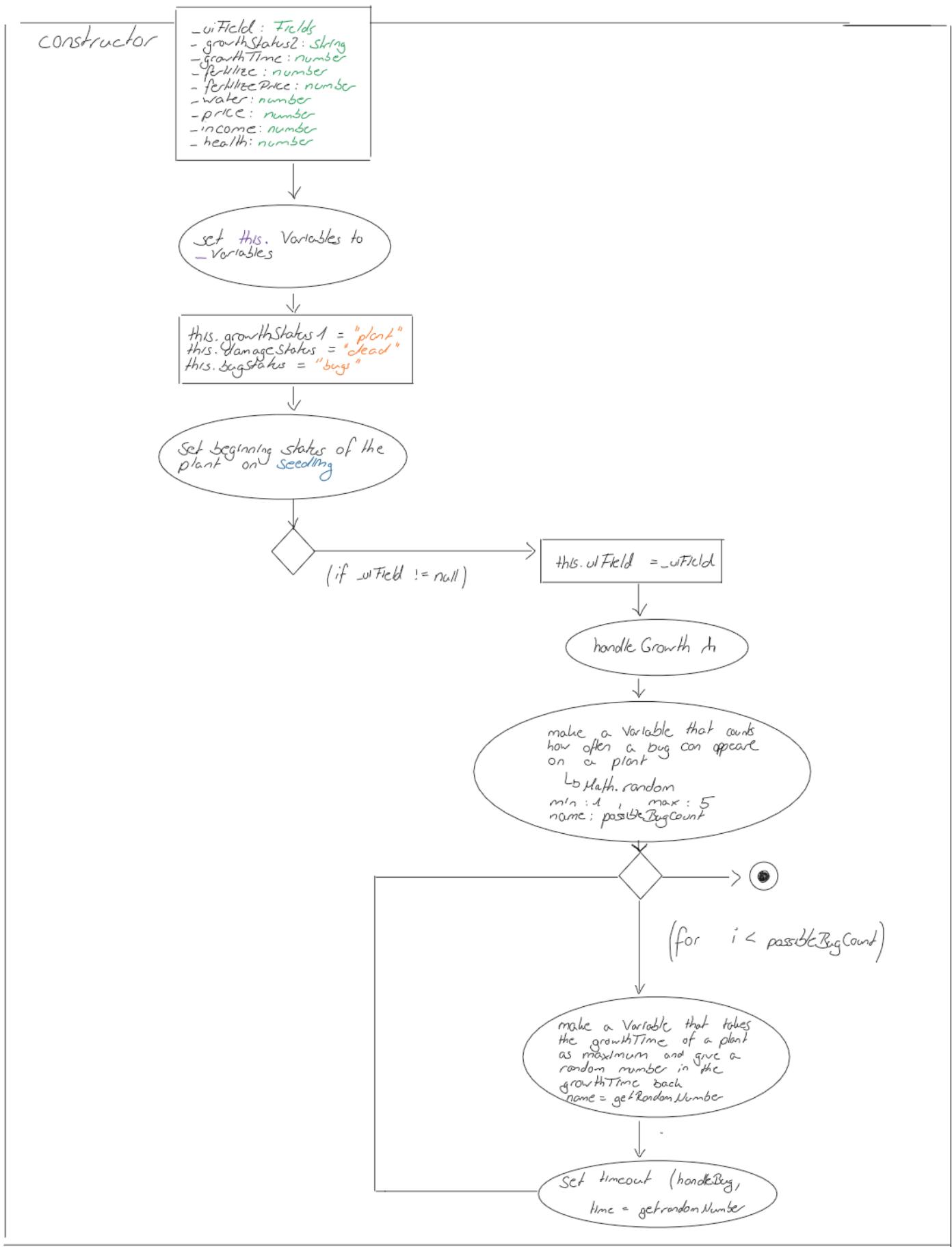
```
gameDiv : HTMLDivElement  
recentVegetable : Vegetable | null  
infoContainer : HTMLDivElement
```

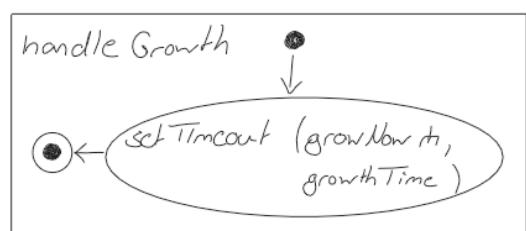
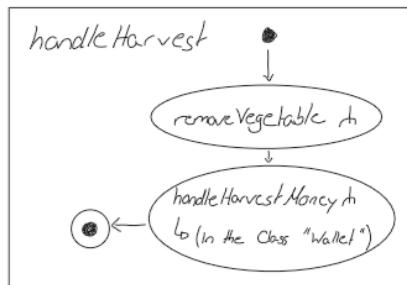
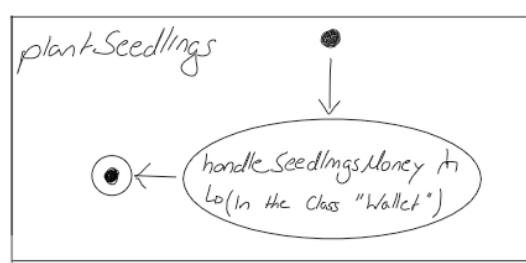
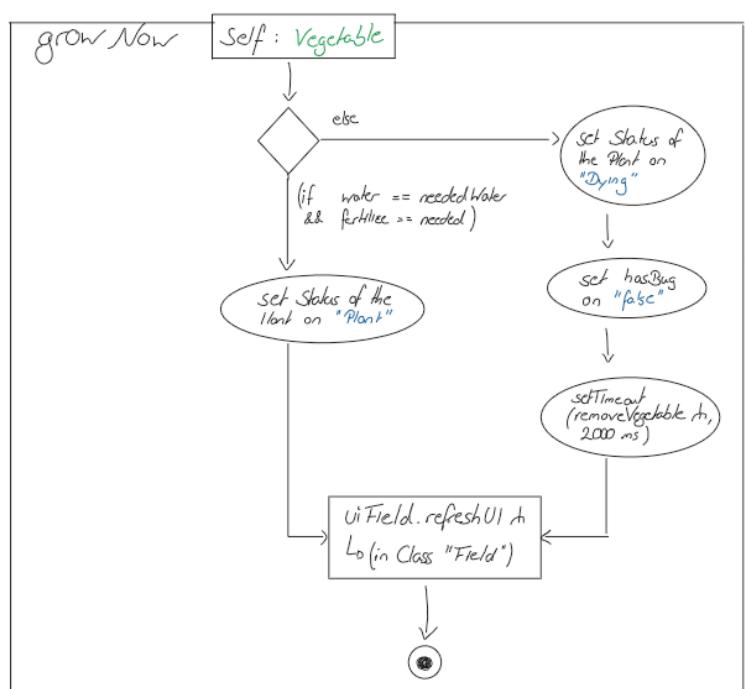
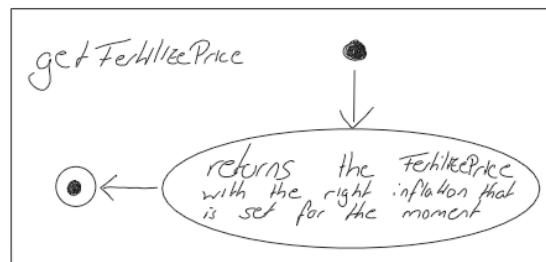
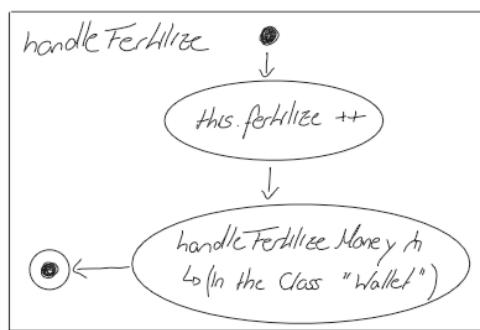
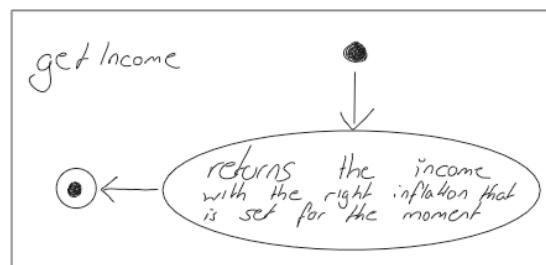
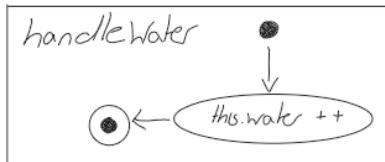
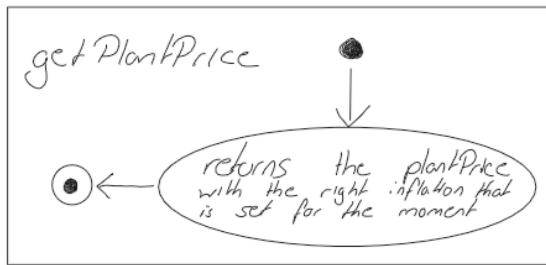


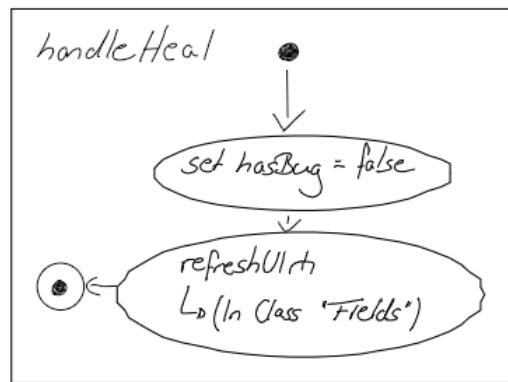
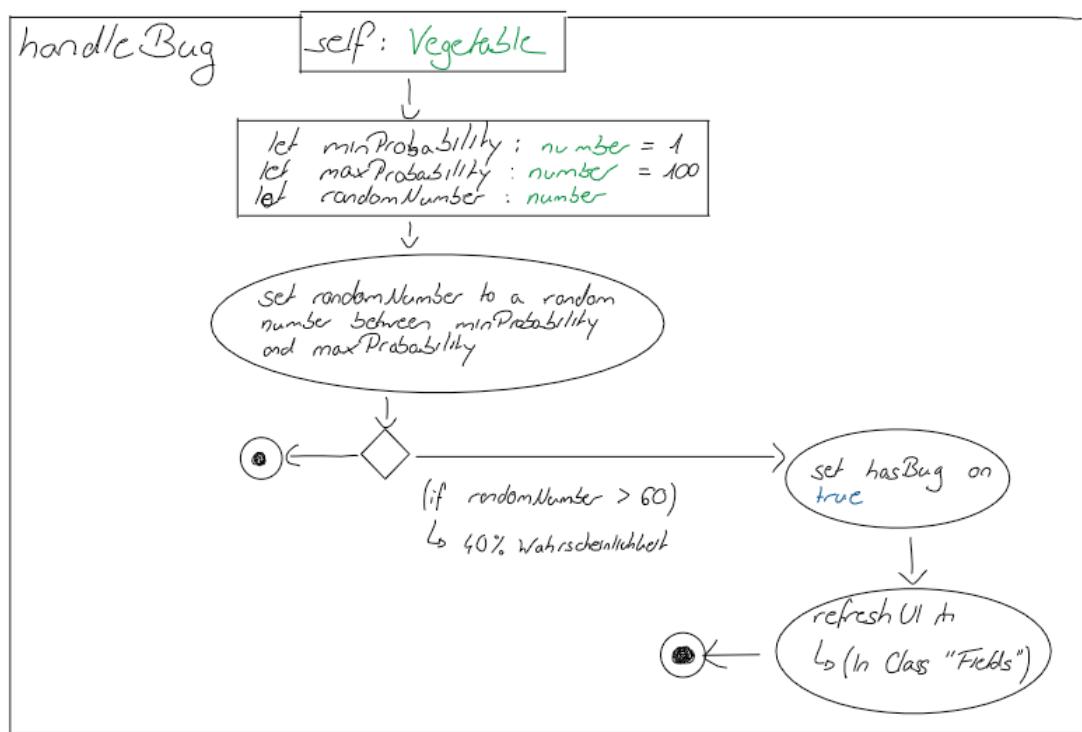
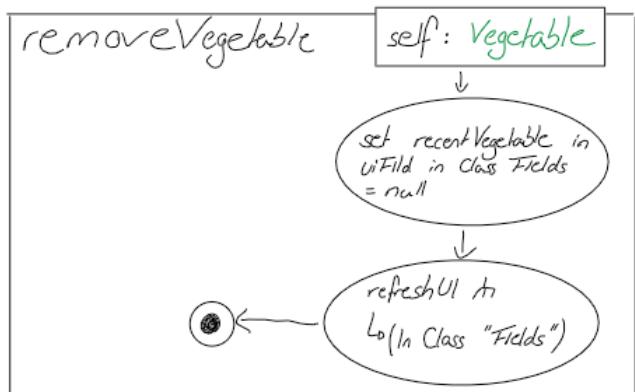




Activity Diagram - Super Class Vegetable

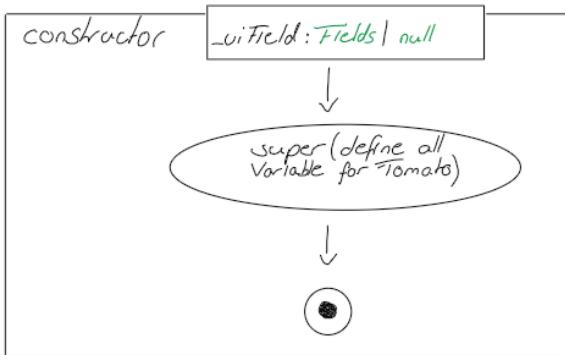






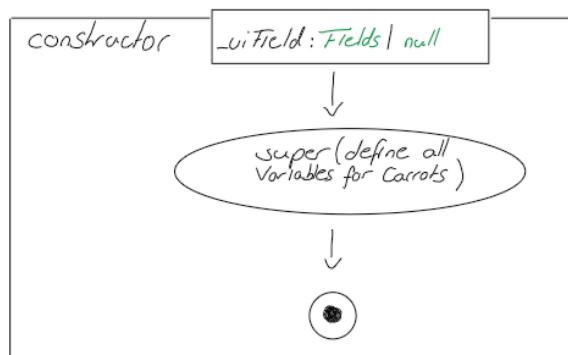
Subclass - Tomato

Static information instance: Tomato = new Tomato(null)



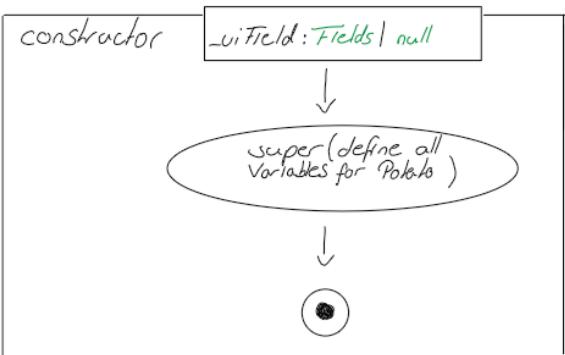
Subclass - Carrots

Static information instance: Carrots = new Carrots(null)



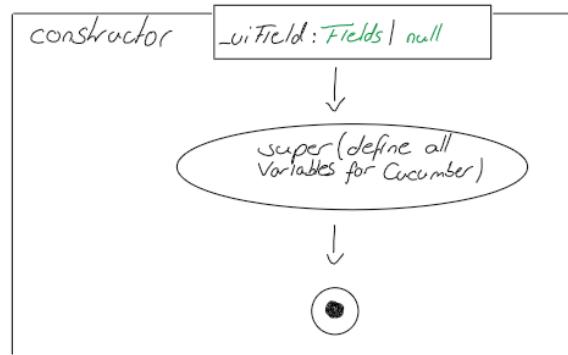
Subclass - Potato

Static information instance: Potato = new Potato(null)



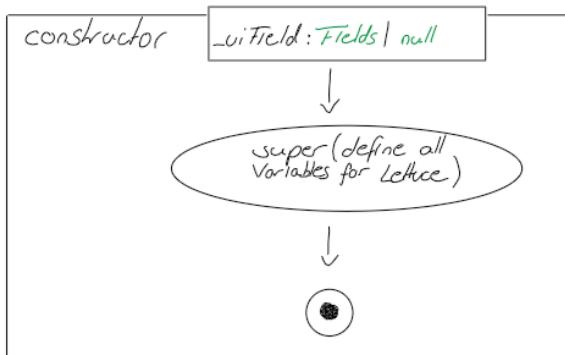
Subclass - Cucumber

Static information instance: Cucumber = new Cucumber(null)



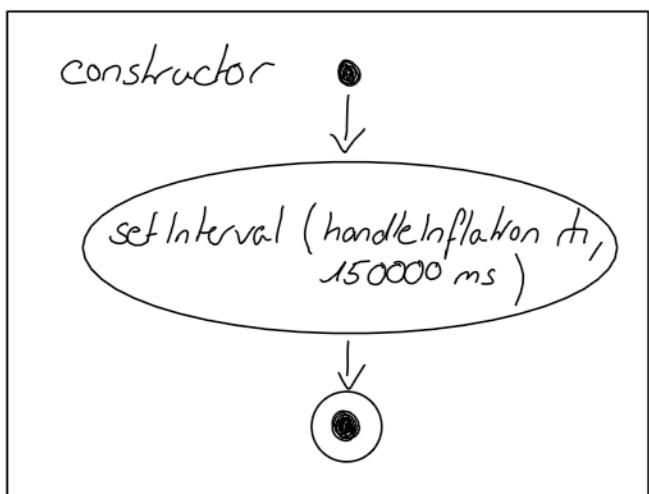
Subclass - Lettuce

Static information instance: Lettuce = new Lettuce(null)



Class - Wallet

```
static instance: Wallet  
walletContainer: HTMLDivElement = document.querySelector  
urlParam: URLSearchParams = new URLSearchParams(window.location.search)  
minPrice: number = get value from Form Element via urlParam  
maxPrice: number = ""  
seedMoney: number = ""  
inflationRatio: number = 1
```



handleInflation



```
let vegetableButtons: HTMLDivElement  
let plantActionButtons: HTMLDivElement
```



get vegetableButtons Div from HTML
and clear the innerHTML



get plantActionButtons Div from HTML
and clear the innerHTML



make a new Variable that gets
it's definition from another function

↳ randomInt from Interval (min, max) →

name = changedInflation



```
inflationRatio = changedInflation / 100
```



alert, so the user sees
the difference



