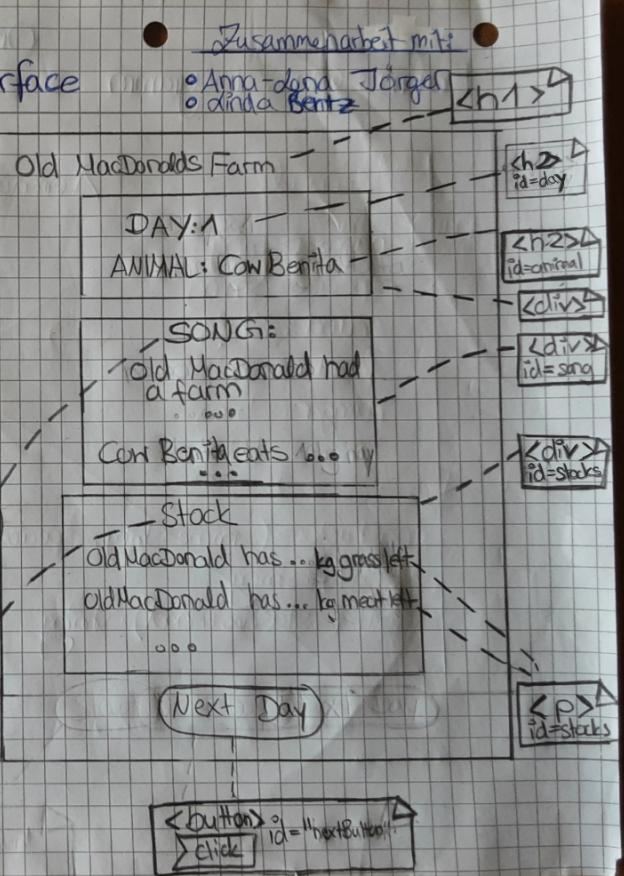
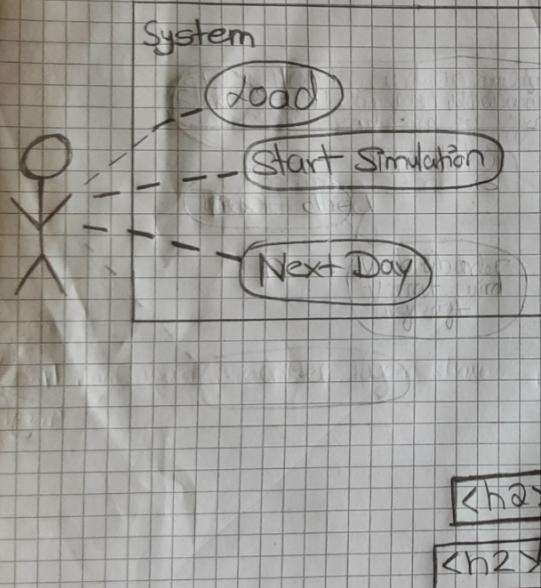


Farm: Use Case Diagram & User Interface

Original

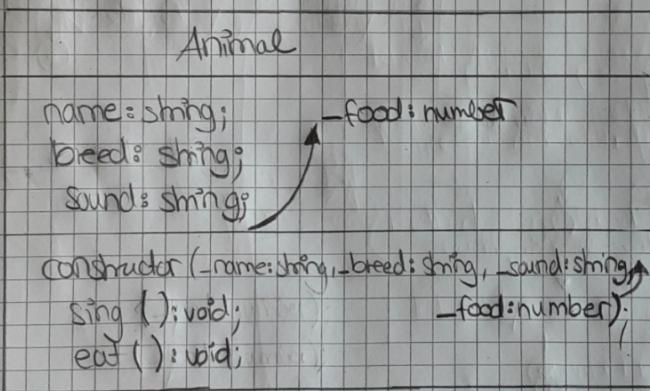


Farm : Class Diagramm

Original

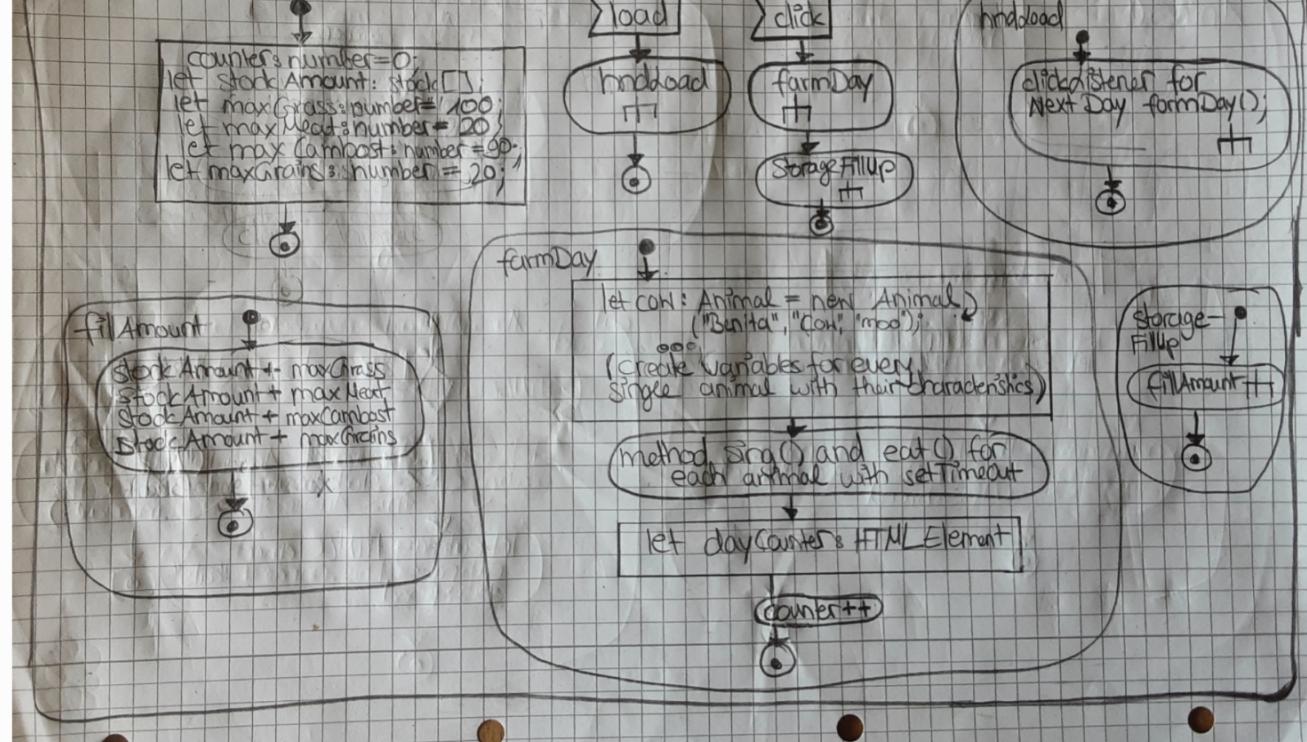
In Zusammenarbeit mit:

- Anna - Linda Jörger
- Linda Bentz

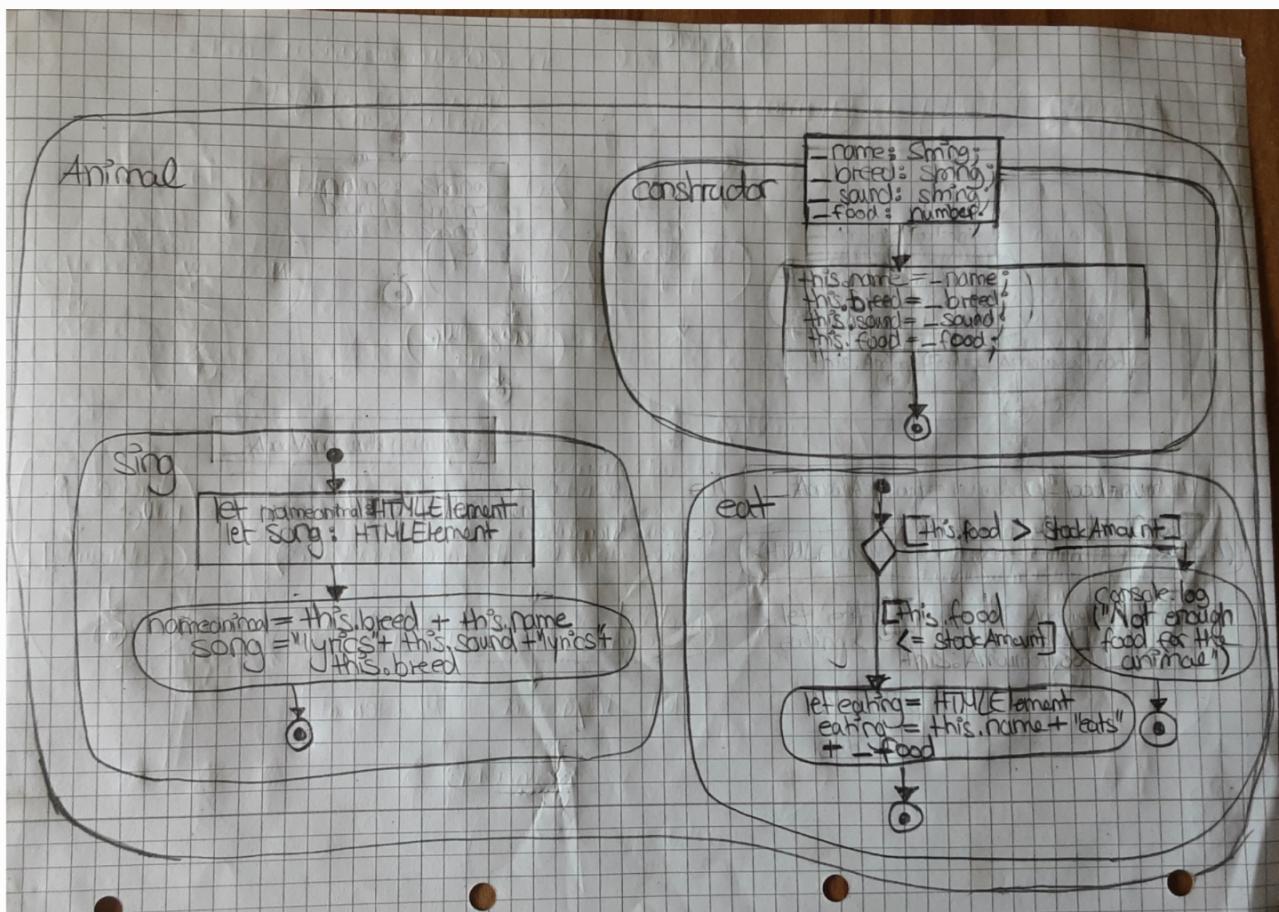


Farm: Activity Diagramm

Farm



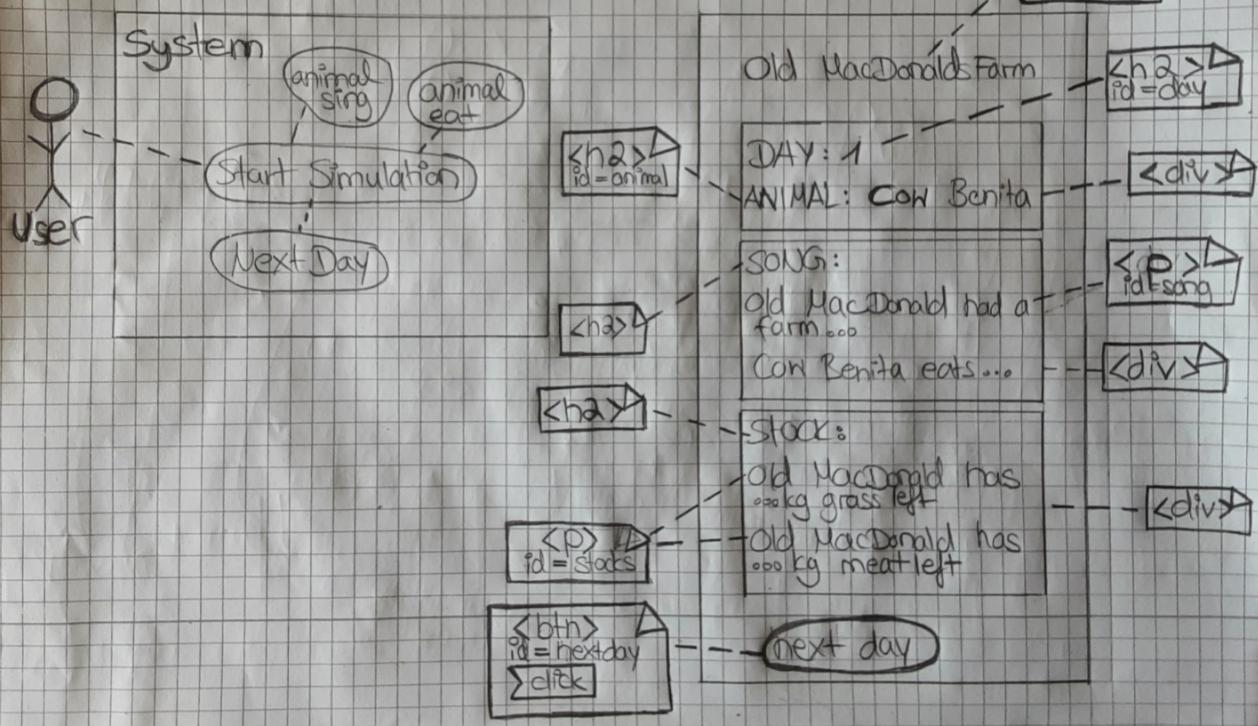
Animal



Farm: Use-Case-Diagramm
& User Interface

Verfasst von:
- Vivien Peedke

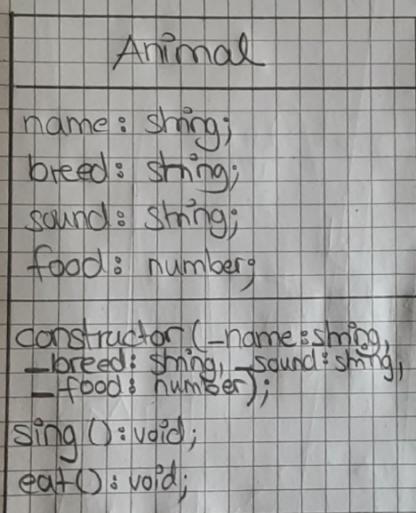
In Zusammenarbeit mit:
- Anna-Lena Jörger
- Linda Bentz



Farm: Class Diagramm

Verfasst von:
- Vivien Peedke

In Zusammenarbeit mit:
- Anna-Lena Jörger
- Linda Bentz



Farm: Activity Diagram

Verbessert von:

- Vivien Peschke

In Zusammenarbeit mit:

- Anna-Dena Jäger

- Linda Bentz

Farm

```
counter: number = 0;
let StockAmount: stock [];
let maxGrass: number = 160;
let maxMeat: number = 20;
let maxCompost: number = 100;
let maxGrains: number = 20;
```



hollLoad

clickListener for
nextDay farmDays();

fillAmount

```
StockAmount + MaxGrass  
StockAmount + MaxMeat  
StockAmount + MaxCompost  
StockAmount -- MaxGrains
```

farmDay

```
let cow: new Animal("Benita",  
"cow", "moo");  
... (create variables  
for every single animal,  
with their characteristics)
```

method sing() and eat()
for each animal with
set meat

Animal

constructor

```
Name: string;  
breed: string;  
sound: string;  
food: number;
```

```
this.name = name;  
this.breed = breed;  
this.sound = sound;  
this.food = food;
```

sing

```
let nameAnimal: HTMLElement  
let Song: HTMLAudioElement
```

```
nameAnimal = this.breed + this.name  
Song = "lyrics" + this.sound +  
"lyrics" + this.breed
```

eat

[this.food > stockAmount]

[this.food < stockAmount]

let eating = HTMLElement

eating = this.name + "eats"
+ food

console.log
(not enough
food for the
animal")