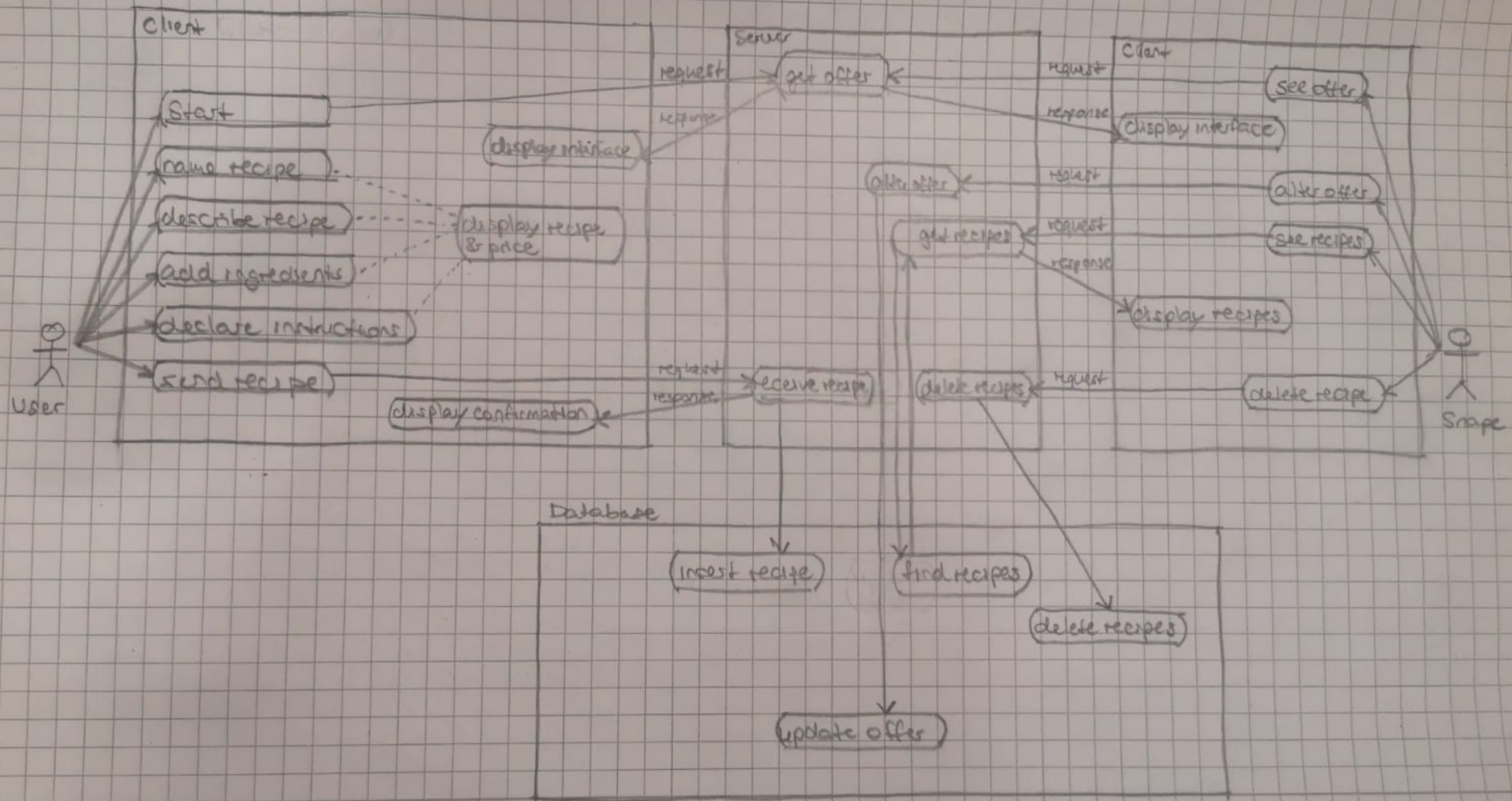


107. Database Hexenkessel: Use Case Diagram



- + LOS Client
- + Log-Serve
- + Log-Database

Cauldron - Recipe

Choose and type in before boiling

Type in / Choose here:

- Name: Type in potion's name
- Description, rise, side effects: Type in here
- Kind of potion: Choose wanted effects
- Duration to full effect: min 1-120
- Instructions:

	amount
Add	°C
Heat to	°C
Cool down to	
Stir	times a minute
- Consistency:
 - ☐ tough
 - ☐ partly
 - ☐ liquid
- Colour:

Your recipe:

- Name
- Description
- Effects
- Instructions
- Amount

Total:

Buttons: Clear recipe, Send recipe, Get recipe

```

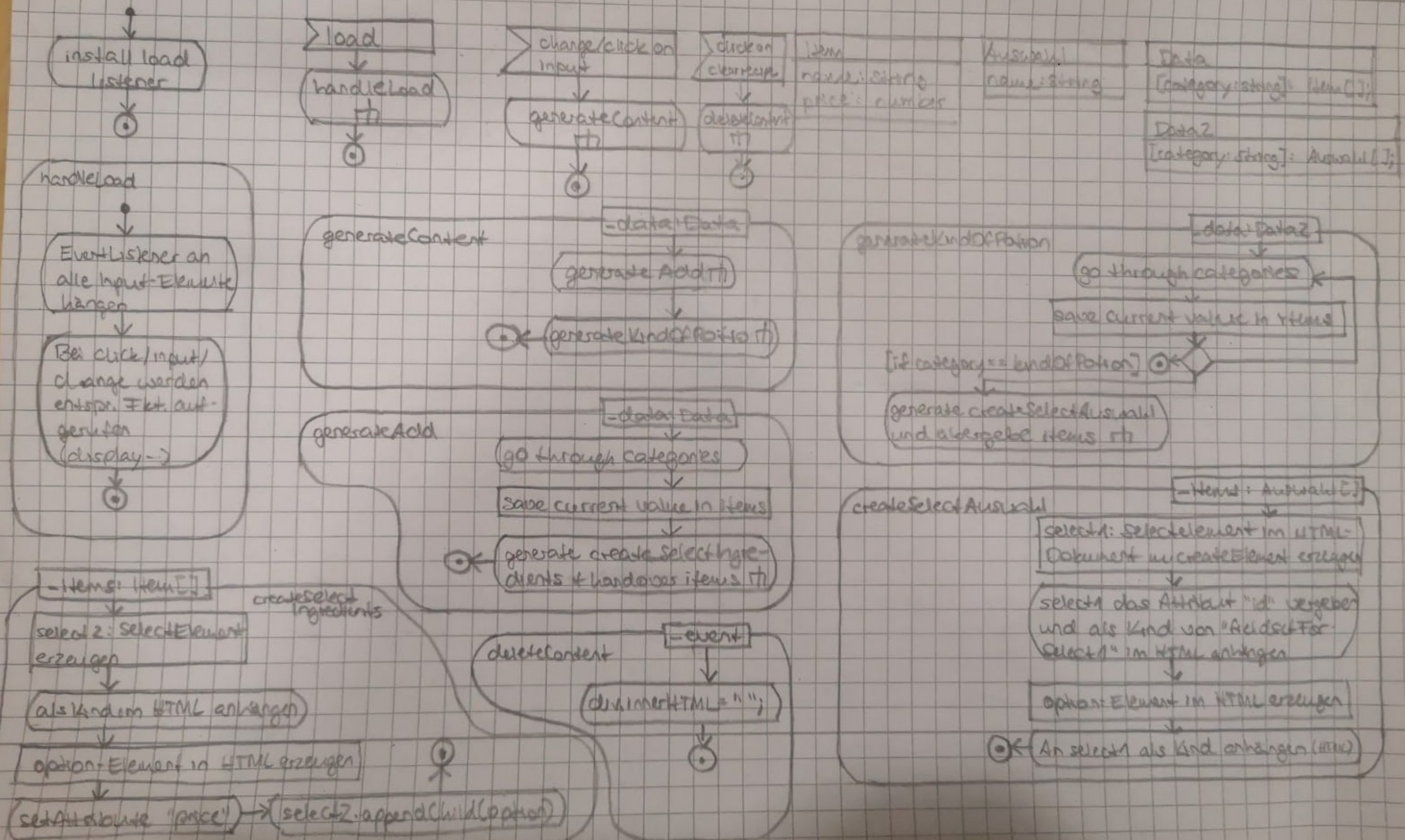
    <p>
    <div id="form">
      <change>
    </div>
    <h1>
    <h2>
    <div id="recipe">
    <p>
  
```

Annotations:

- <pre></pre>
- <input type="text">
- <textarea>
- Function generates kind of Potion (selects duration)
- <input type="range">
- friendly helper id
- function generates Add select instructions
- <p>
- <input type="range">
- <input type="radio">
- <input type="color">
- <button type="button">
- <input type="number">
- <button type="submit">

LO5 Client Hexenkessel, Activity Diagram +L06 +L07

— hexenkessel.ts —



LOS Client Hexenkessel — defaults —

category	name: string	price: number	data
Ingredients	firefly	1	category
	earwigs	0.4	
	

(JSON)

```
{ ingredients: [ { name: "firefly", price: 1 },  
                 { name: "earwigs", price: 0.4 },  
                 ...  
               ]  
};
```

LO6 - Server Hexenkessel
+ LO7

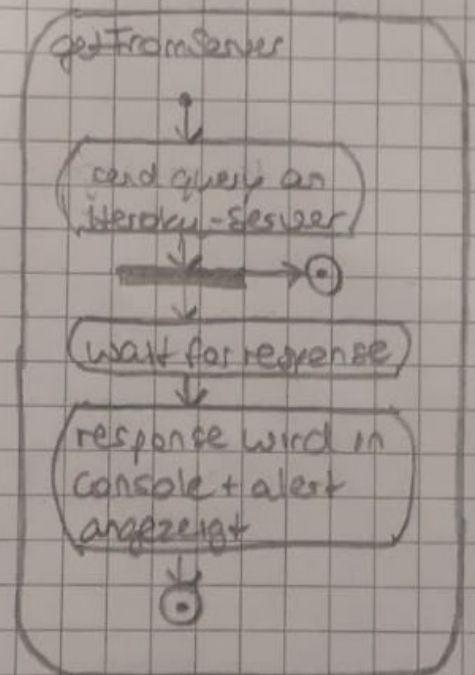
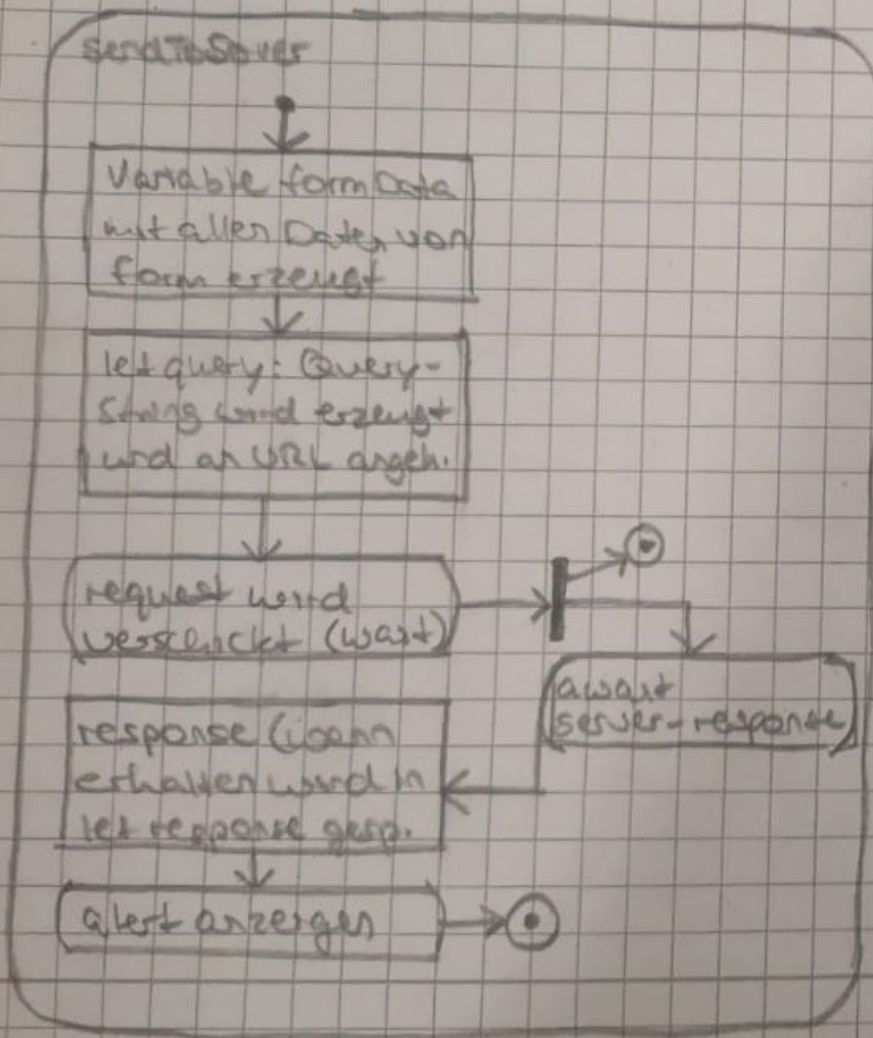
— clientScript.ts —

Click on
send recipe

click on
get recipe

sendToServer
()

getFromServer
()



LOG Server Hexenkessel

— server.ts —

Activity Diagram

