

$v = v + 1$

↑ wird addiert und hier gespeichert
stelle im Speicher

v = variabel

variable wird mit **let** angelegt : let v ;

Definition von v : $v = 1$

Konsequente Verwendung von let

Datentypen

number, string, boolean

→ gewünschte Datentypen schon vorher festlegen

Array (Datenstruktur)

→ alles kann an beliebige Stellen des Arrays „reingeworfen“ werden

Assoziatives Array

→ geschweifte Klammer
→ Schlüssel-Werte-Paare
→ Assoziation :

→ Datentypen heterogen
→ Schlüssel können beliebig gewählt werden

Let $s = \{ "zahl": 7, "wahr": true, text: "Hallo" \}$

$s.zahl \rightarrow 7$

$s["zahl"] \rightarrow 7$

$s[4] = [101, 102] \rightarrow \text{Object} \{ 4: \text{Array} [101, 102], text: "Hallo", wahr: true, ... \}$

interface

- Schlüssel vordefinieren
- Datentypen für Werte + Schlüssel einschränkbar

```
interface MapStringToBoolean {  
    [key: string]: boolean;  
}  
let a: MapStringToBoolean = {"wert1": true, "wert2": false};
```

} Schlüssel frei wählbar,
aber auf Typ String
beschränkt, auf
Wahrheitswerte homogenisiert

```
interface VectorWithMeaning {  
    x: number;  
    y: number;  
    meaning: string;  
}  
let vector: VectorWithMeaning = {x: 12.4, y: -7.2, meaning: "Ortsvektor"};
```

Function

- Code innerhalb eines Programms, das ein eigenes kleines Programm erstellt

```
function name (parameter)  
{ Funktionskörper }
```

Modulo : teilbar ohne Rest → true

return führt dazu, dass die Funktion an der Stelle verlassen wird

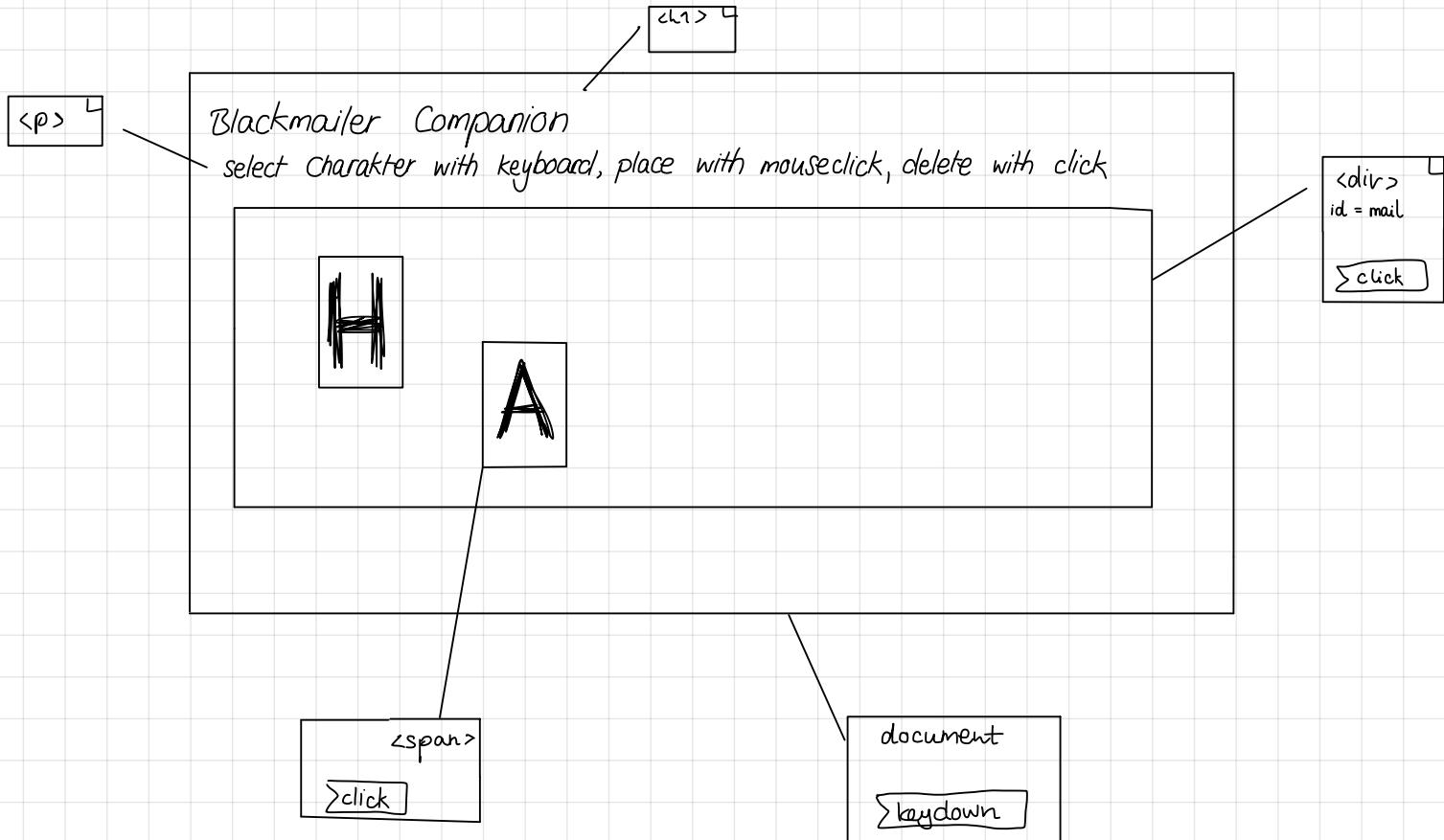
Objekt

- assoziatives Array an dem Funktionen anhaften

debugger

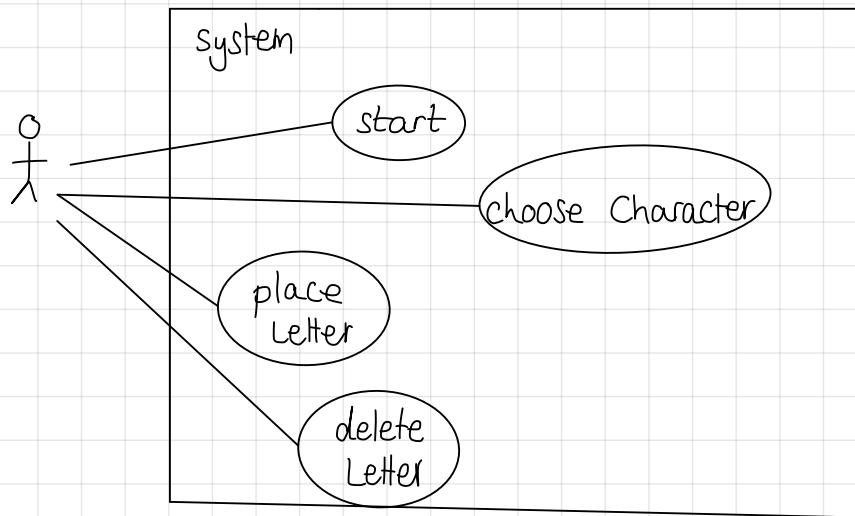
L02 Events UI - Scribble

Blackmailer Companion : UI - Scribble

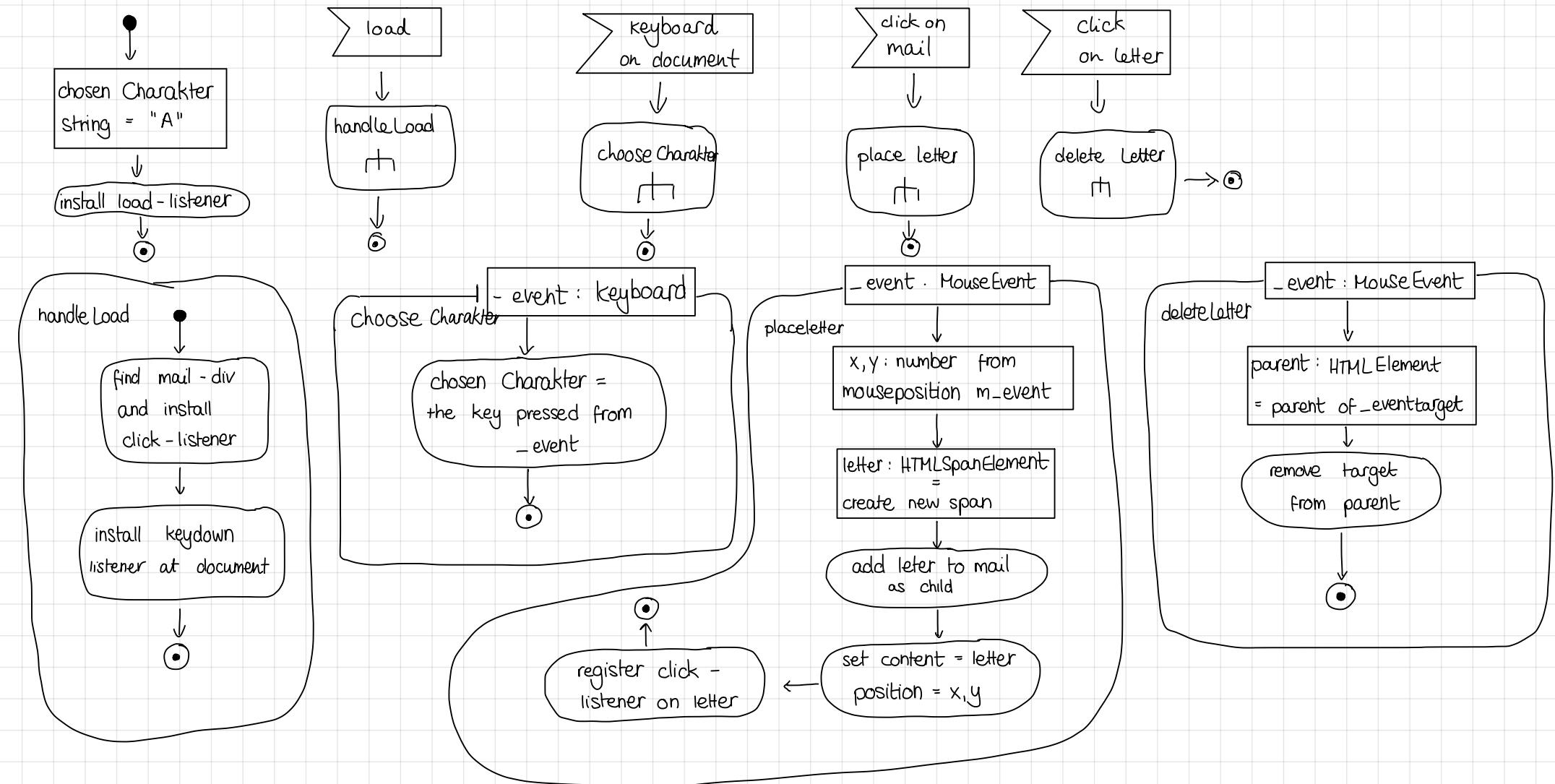


L02 Events

Blackmailer Companion : Use - Case - Diagramm (Anwendungsfalldiagramm)



Blackmailer Companion : Activity Diagram



Blackmail Companion

```
namespace LOZ-BlackmailerCompanion {
    console.log("Start")
    let chosenCharacter: string = "A";
    window.addEventListener("load", handleLoad);

    function handleLoad(_event: Event): void {
        let mail: HTMLElement = <HTMLElement> document.querySelector("div#mail");
        mail.addEventListener("click", placeLetter);
        document.addEventListener("keydown", chooseCharacter);
    }

    function placeLetter(_event: MouseEvent): void {
        console.log(_event);
        let x: number = _event.offsetX;
        let y: number = _event.offsetY;

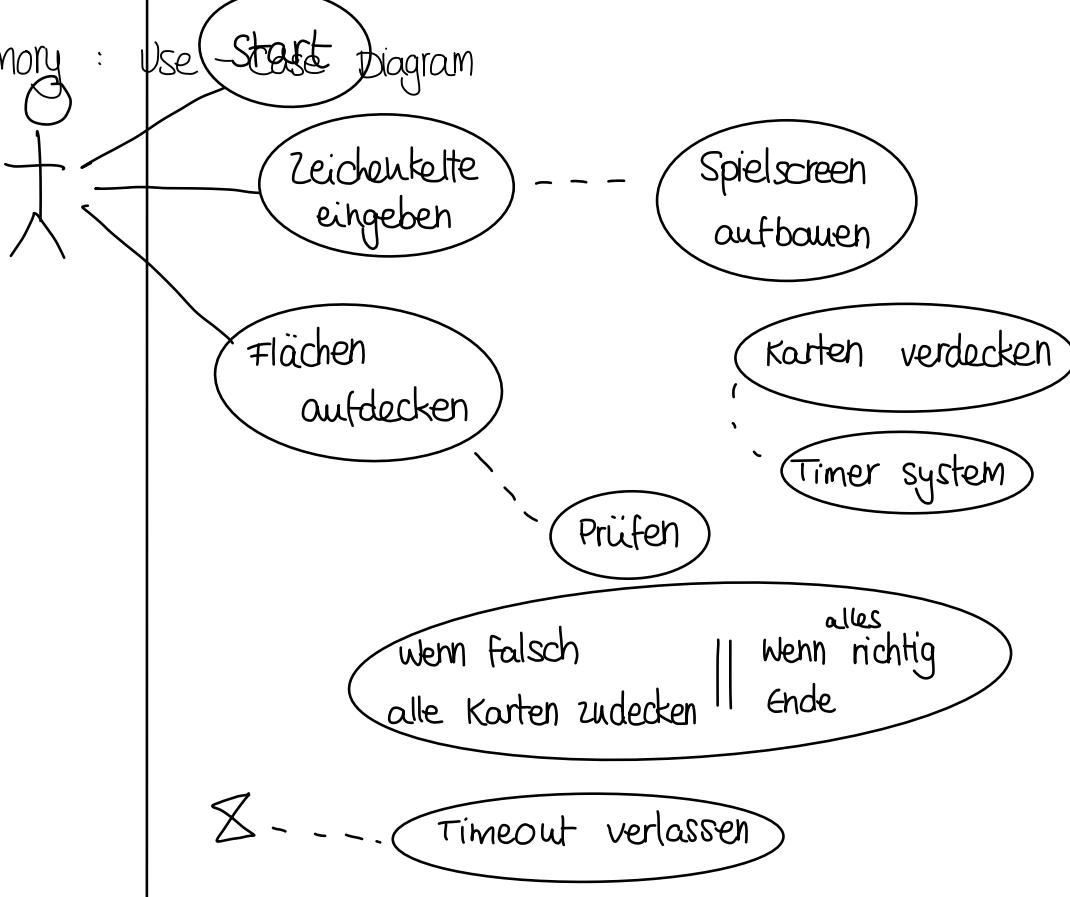
        let mail: HTMLElement = <HTMLElement> _event.target;
        let letter: HTMLSpanElement = document.createElement("span");
        mail.appendChild(letter)

        letter.textContent = chosenCharacter;
        letter.style.left = x + "px";
        letter.style.top = y + "px";
    }

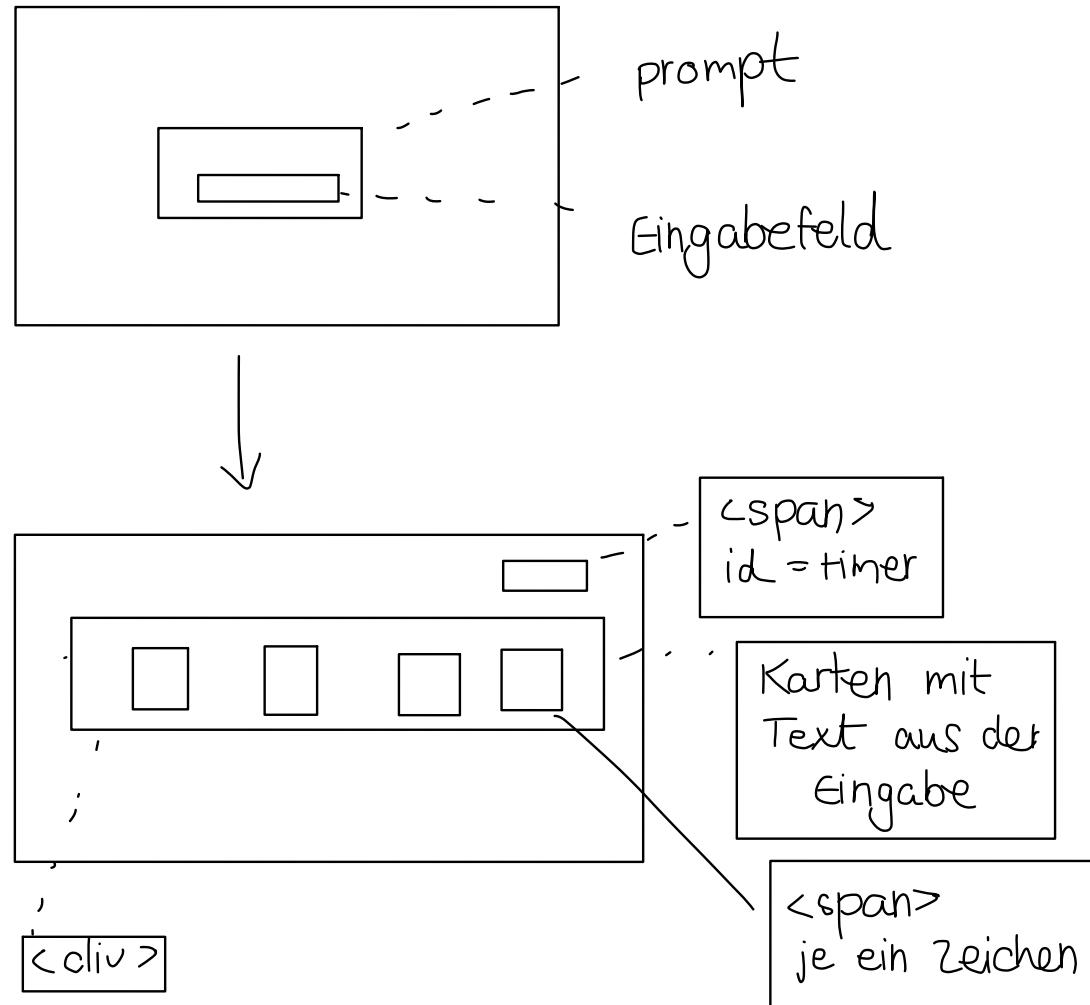
    function chooseCharacter(_event: KeyboardEvent): void {
        //console.log(_event);
        chosenCharacter = _event.key
    }

    function deleteLetter(_event: MouseEvent): void {
        let target: Node = <Node> _event.target;
        let parent: Node = <Node> target.parentNode;
        parent.removeChild(target)
    }
}
```

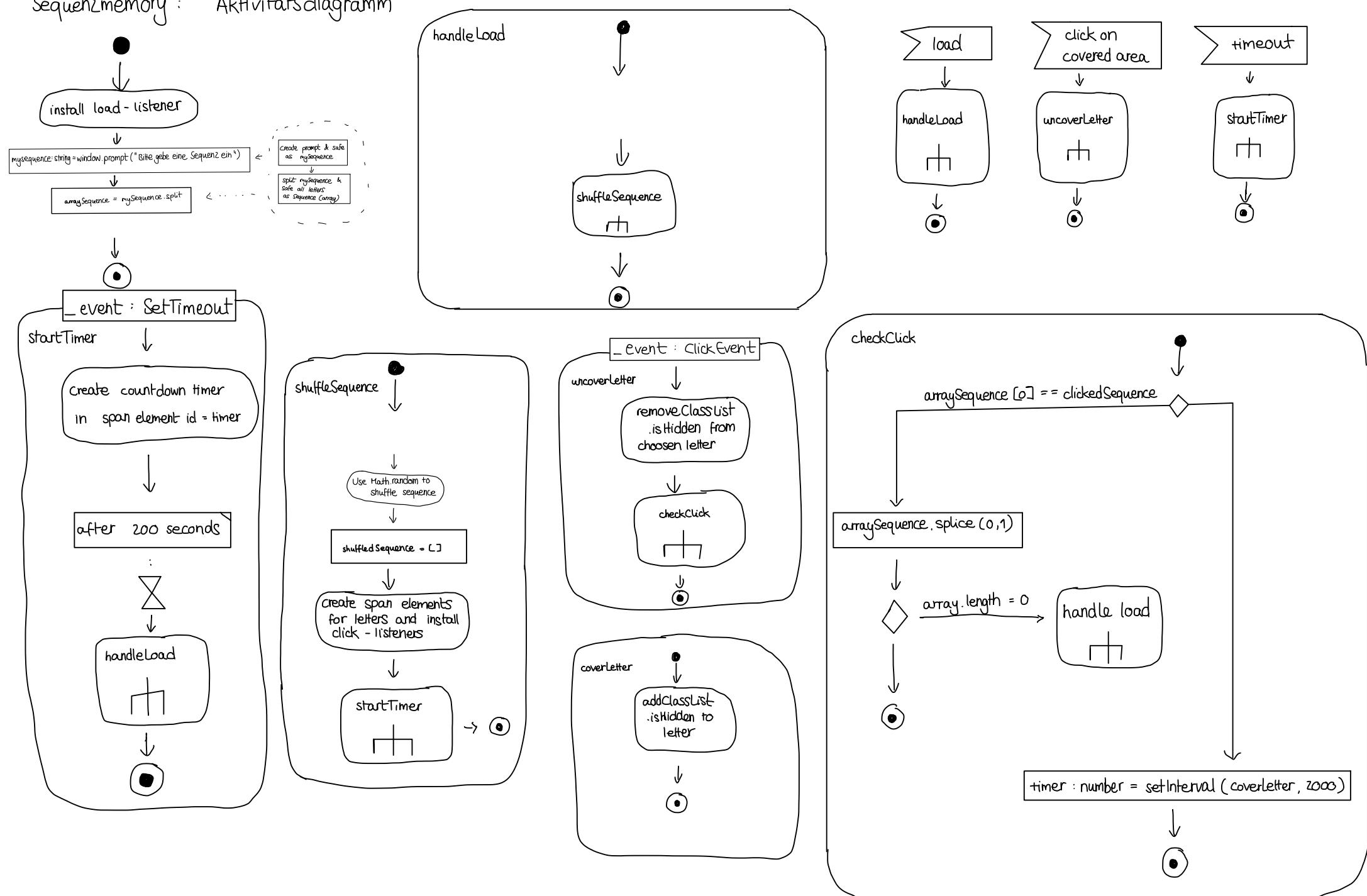
Sequenz-Memory : Use Start State Diagram



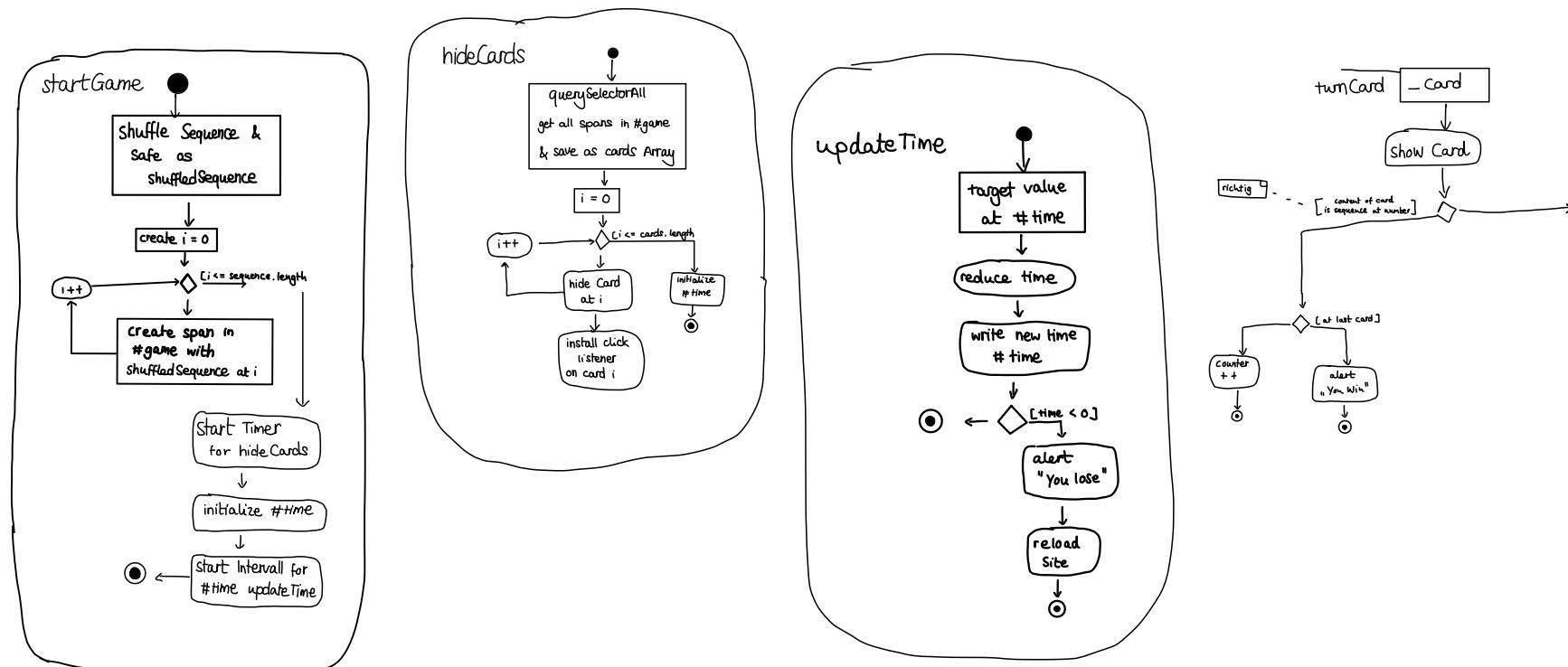
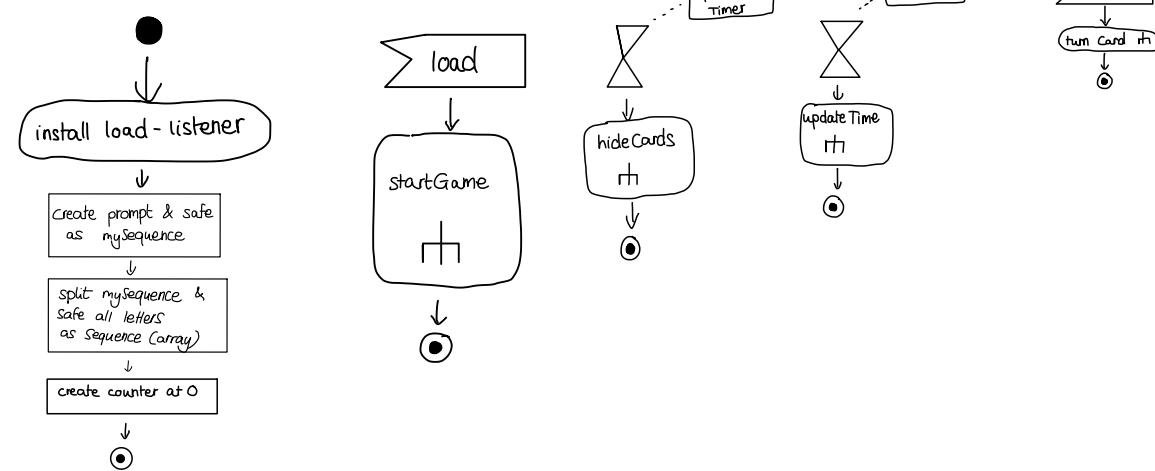
UI - Scribble



Sequenzmemory : Aktivitätsdiagramm



Sequenzmemory : Aktivitätsdiagramm



In meinem Spiel zuerst string let string = ""
let stringArray = string.split(" ");

Array.prototype.memory_tile_shuffle = funktion() {

Karten
Mischen
var i = this.length, j, temp;
while (--i > 0) {
 j = Math.floor(Math.random() * (i+1));
 temp = this[j];
 this[j] = this[i];
 this[i] = temp;

Karten gedreht

function newBoard()
 tiles_flipped = 0;
 let output = '';
 memory_array.memory_tile_shuffle();

for (var i = 0; i < memory_array.length; i++) {
 output += '

'

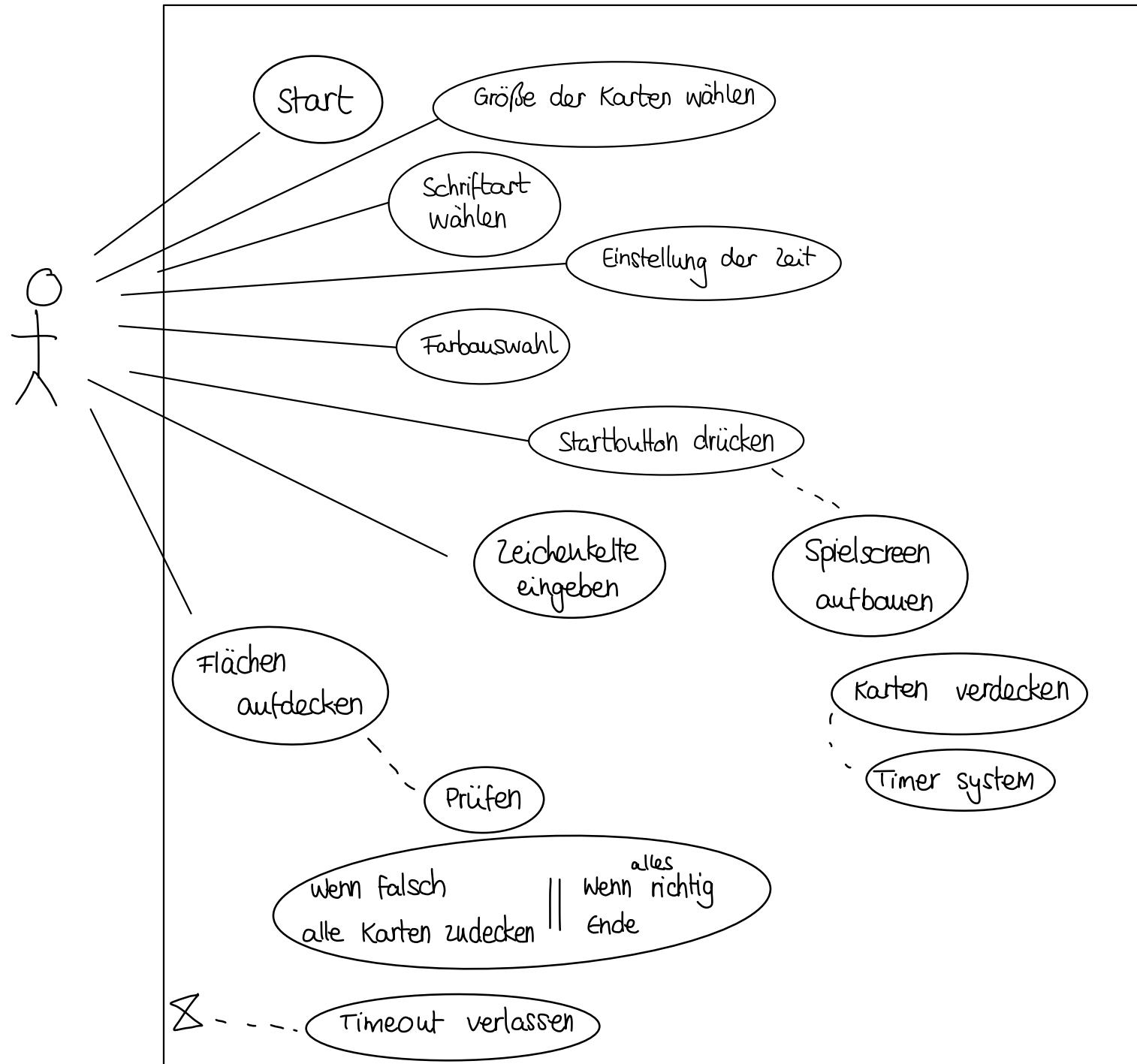
document.getElementById('memory_board').innerHTML = output;

window.addEventListener(newBoard)

memoryFlipTile (cart, val)

tile.style.background = '#FFF'

Sequenzmemory : Use - Case - Diagram



SequenceMemory Settings

Sequence

colorpicker

background

cards

font

Size Cards

small

big

Fonts

o

o

o

Time

17

start

Klausurtermine :

EIA2

PW: MingoDB

2/3 Punktzahl Klausuren 45 min

- Mühdl. Prüfung : - Ampel n. rot
- immer was abgeben
- letzte 2 Tage Prüfungszeit
- Code habwegen funktioniert

EIA2 L01 Recap + Foundation

NaN = Not a Number

let / const name : type = value;

Primitive Datentypen

- number
- string
- boolean

Komplexe Datentypen

- Array : let a: number [] = [7, true, "Hallo"]
- Assoziatives Array : let s = { "zahl": 7, "wahr": true, "text": "Hallo" }
- Interface : interface Student {
 name: string;
 matrikel: number;
 grades: [module: string]: number;
}
let students: Student [] = [];
students.push({ name: "Big Brain", matrikel: 123456, grades: { "EIA1": 1.3 } })

L01 Tracetable

COWS

nums: number [] = [2, 6, 5]

main

Nr	results string []	i number	result string	Kommentar
3	[]			
4		0	"muhu"	$0 < 3 ? \rightarrow \text{true}$ create call
5				$i < 3 ? \rightarrow \text{true}$ create call
6	["muhu"]	1		
4				
5				

create Call				
Nr	- Start String	- length number	K number	Kommentare
10	"m"	2	2	$K > 0 ? \rightarrow \text{true}$ if $\rightarrow \text{false}$
11				
12				
14	"mu"	1	1	$1 > 0 ? \rightarrow \text{true}$ if ($K == 1$) $\rightarrow \text{true}$
11				
12				
13	"muh"	0	0	$0 > 0 ? \rightarrow \text{false}$
14	"muhu"			
11				
16				
10	" m "	6	6	$6 > 0 ? \rightarrow \text{true}$ if $\rightarrow \text{false}$
11				
14	" mu "	5	5	$5 > 0 ? \rightarrow \text{true}$ if $\rightarrow \text{false}$
11				
14	" muu "	4	4	$4 > 0 ? \rightarrow \text{true}$
11				
14	" muuu "	3	3	$3 > 0 ? \rightarrow \text{true}$ $3 == 3 \rightarrow \text{true}$
11				
13	" muuh "			
14	" muuhu "			

L01_ Zufallsgedicht

Words

Tracetable

subjekte: string [] = ["Harry", "Hermine", "Ron", "Hagrid", "Snape", "Dumbledore"]

praedikate: string [] = ["braut", "liebt", "studiert", "hasst", "zaubert", "zerstrt"]

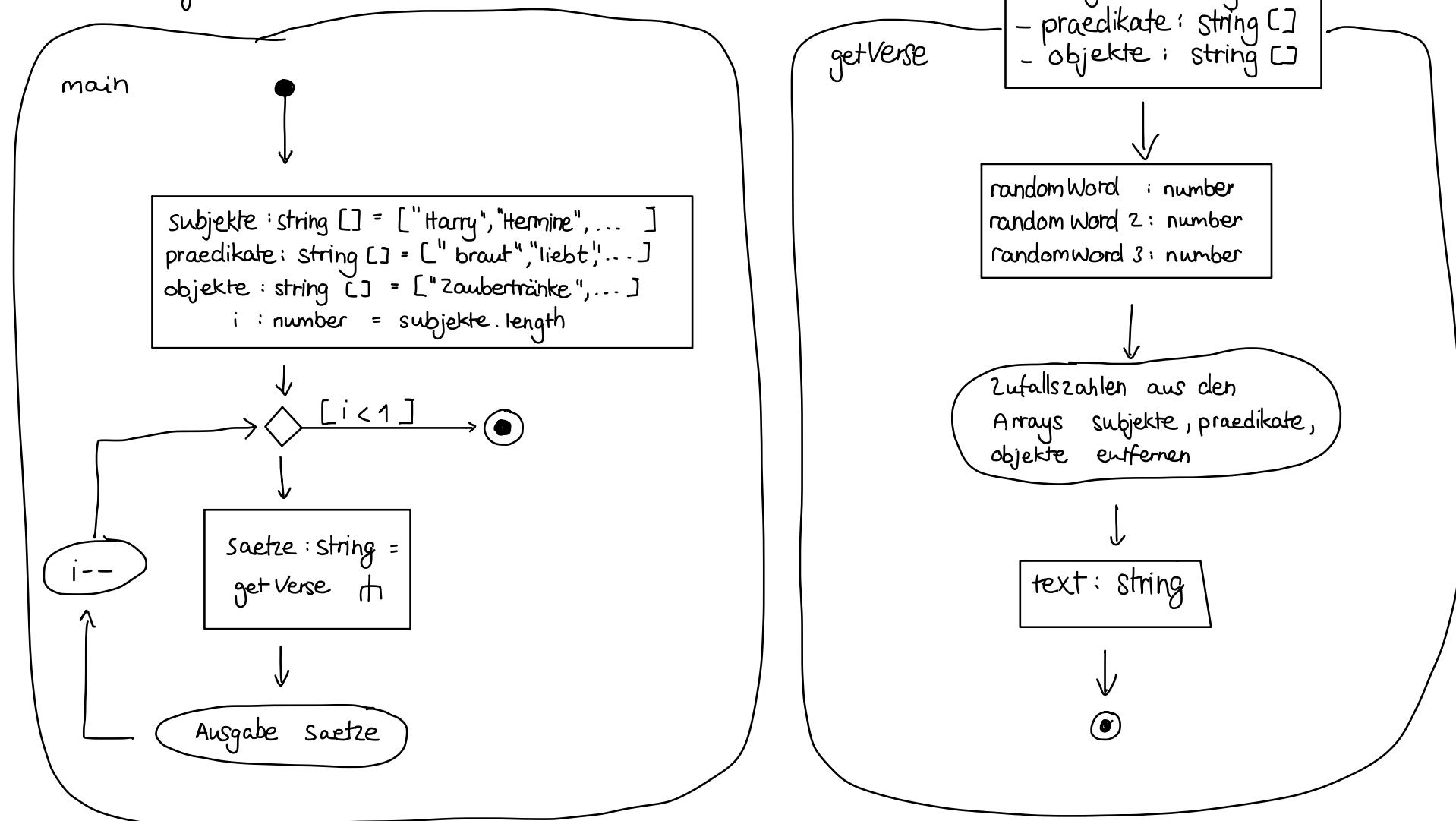
objekte : string [] = ["Zaubertrnke", "den Grimm", "Lupin", "Hogwarts", "die Karte des Rumtreibers", "Dementoren"]

main

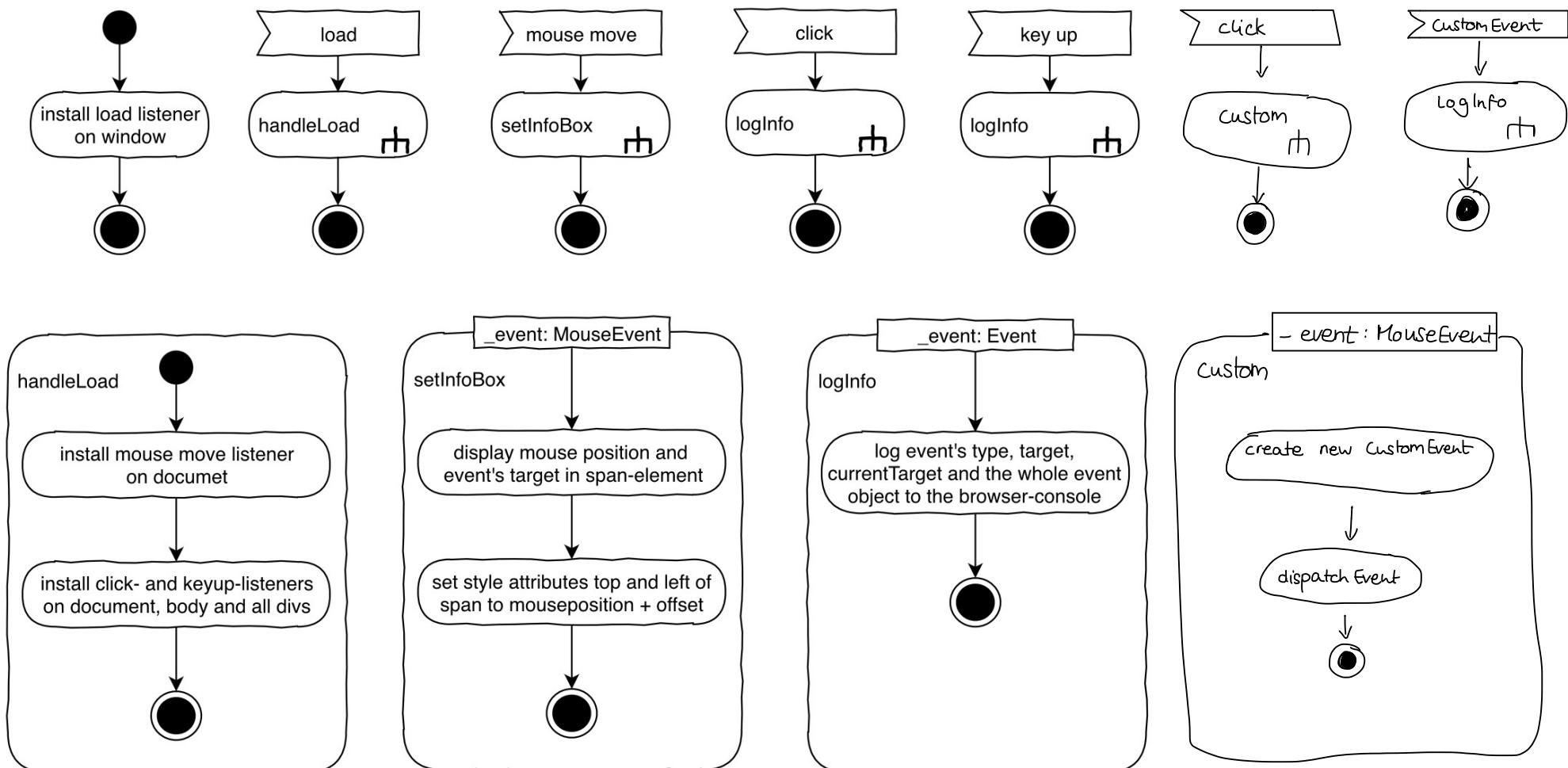
Nr	number	saetze string	Kommentar	getVerse				text string	Kommentar
				Nr	random Word number	random Word 2 number	random Word 3 number		
23 27	6	"Harry zaubert Hogwarts"	$6 \geq 1$ get Verse	35 36 37 38	0 4 3				
23 27	5	"Hermine liebt Lupin"	$5 \geq 1$ get Verse	35 36 37 38	0 1 2			"Harry zaubert Hogwarts"	
	:			35 36 37 38					
	:			35 36 37 38					
	:			35 36 37 38					
	:			35 36 37 38					
	:			35 36 37 38					
23 27	1	"Ron braut Zauber- trnke"	$1 \geq 1$ get Verse	35 36 37 38	0 0 0				"Ron braut Zaubertrnke"

L01_Zufallsgedicht

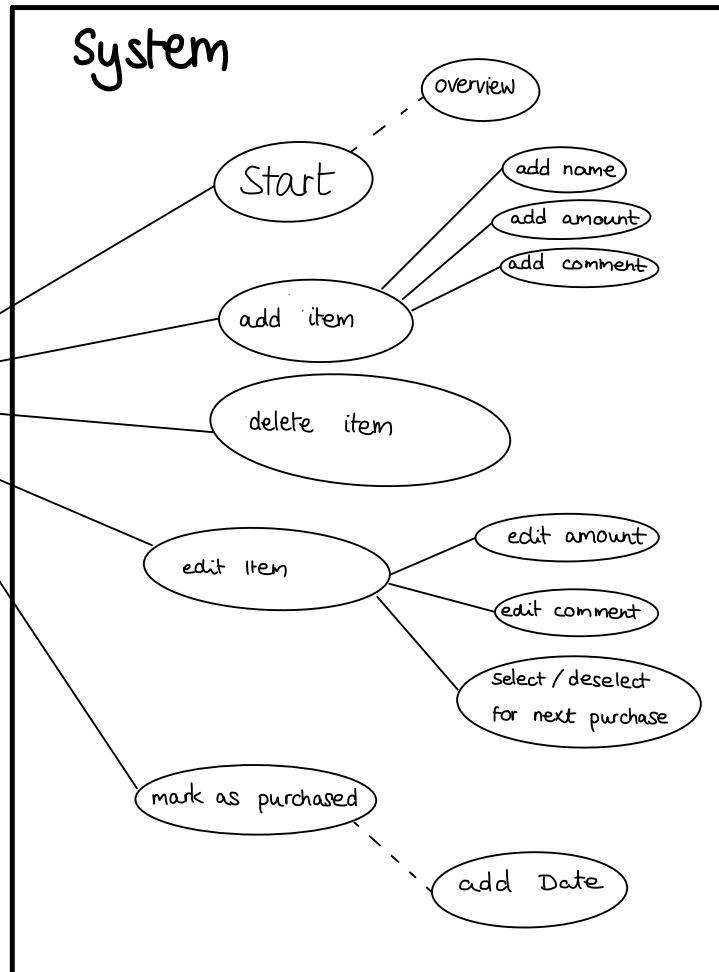
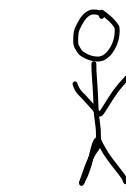
Aktivitätsdiagramm



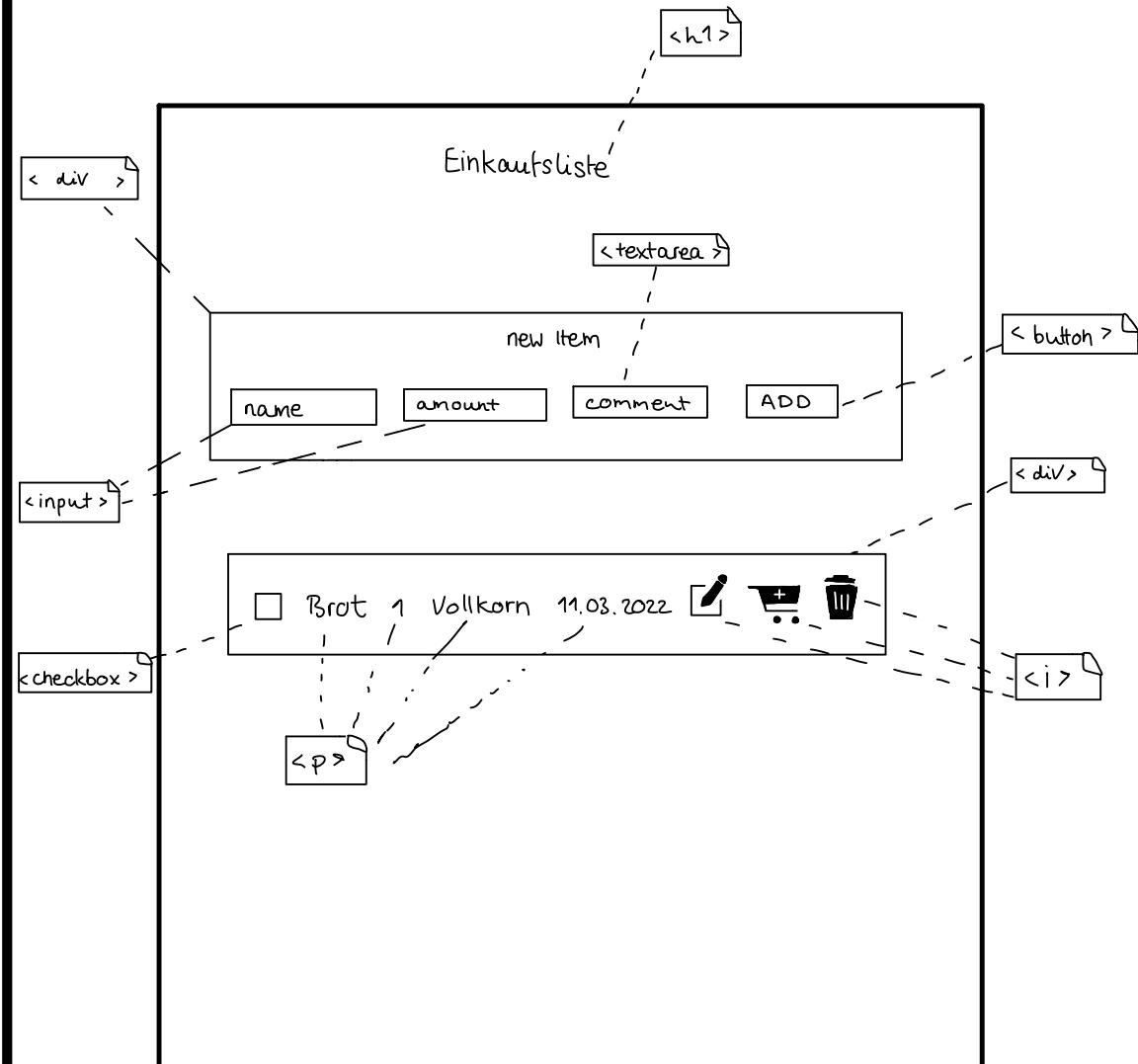
Event-Inspector: Activity-Diagram



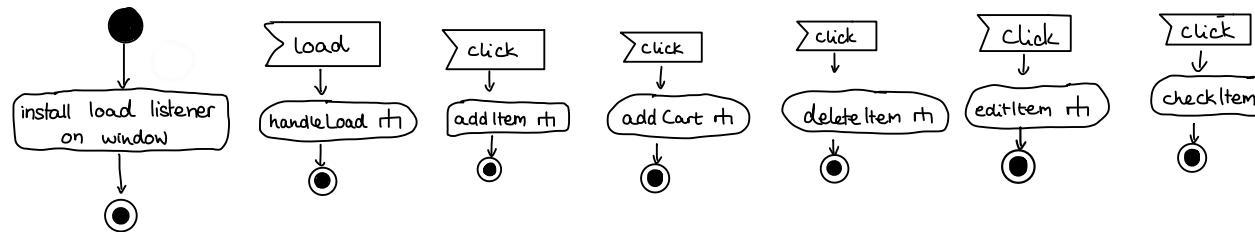
ShoppingList · Use-Case - Diagramm



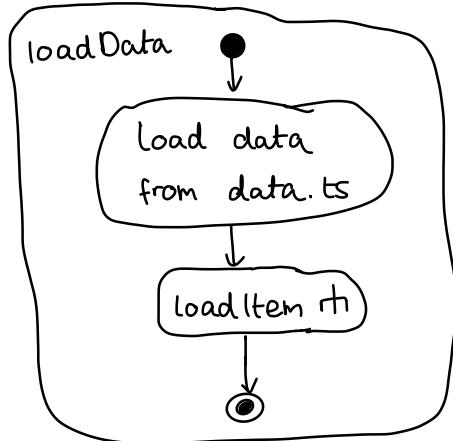
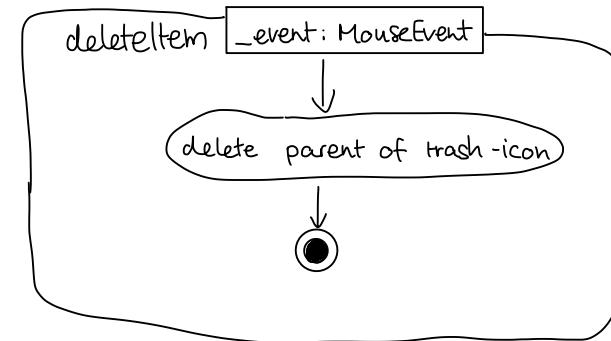
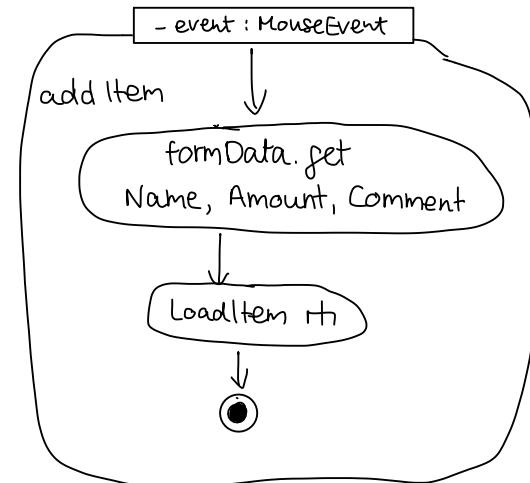
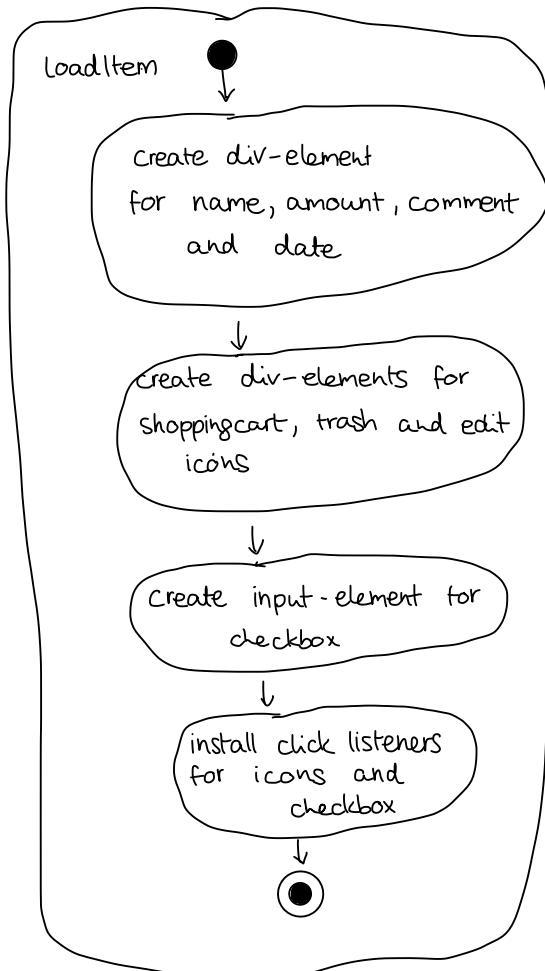
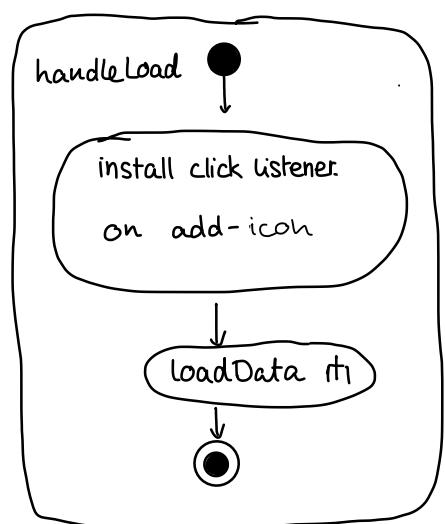
ShoppingList · UI - Scribble



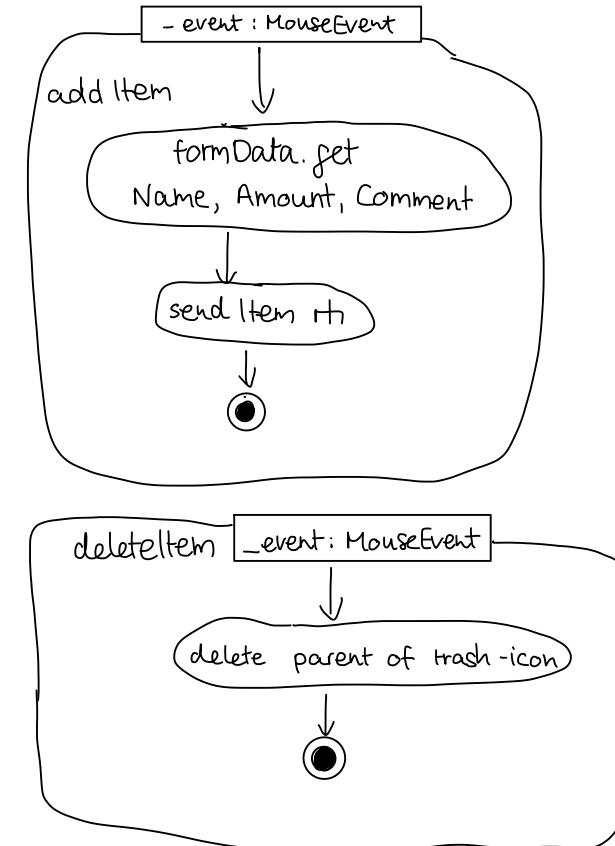
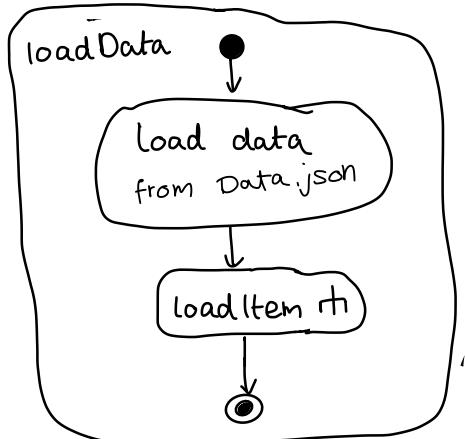
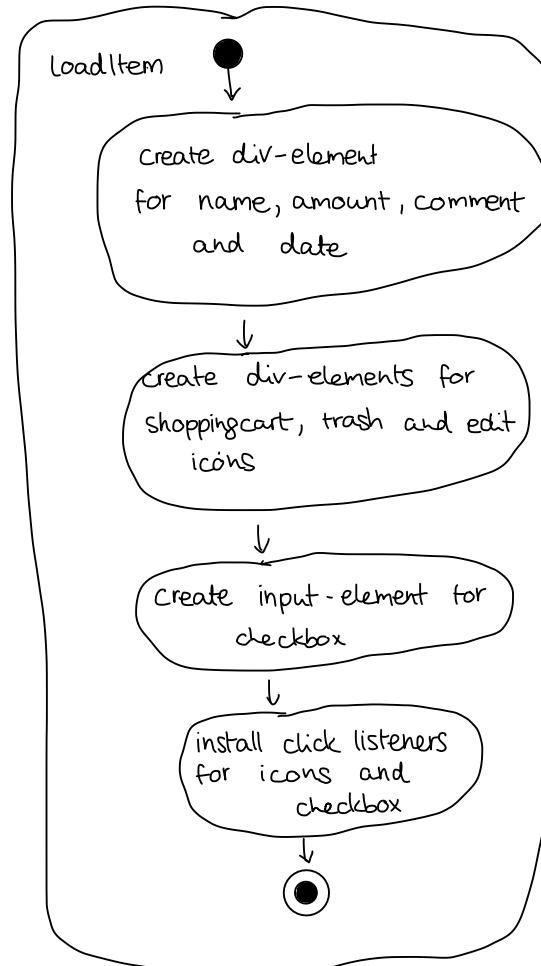
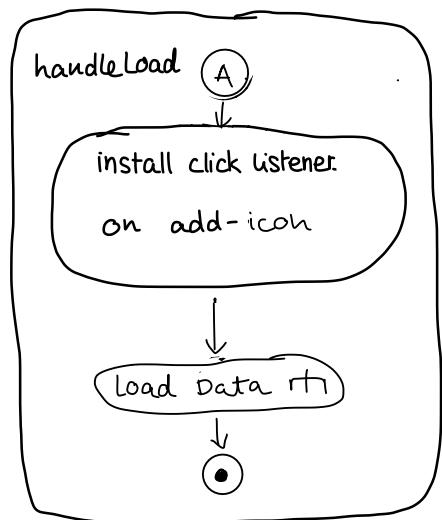
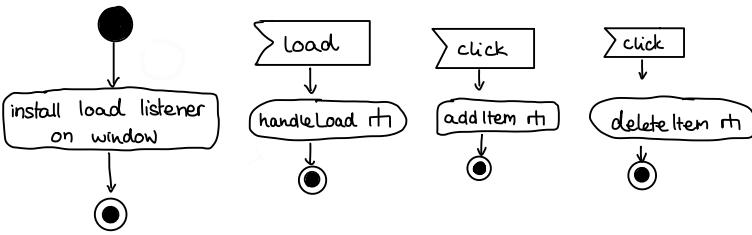
L04 - ShoppingList - DataStructure



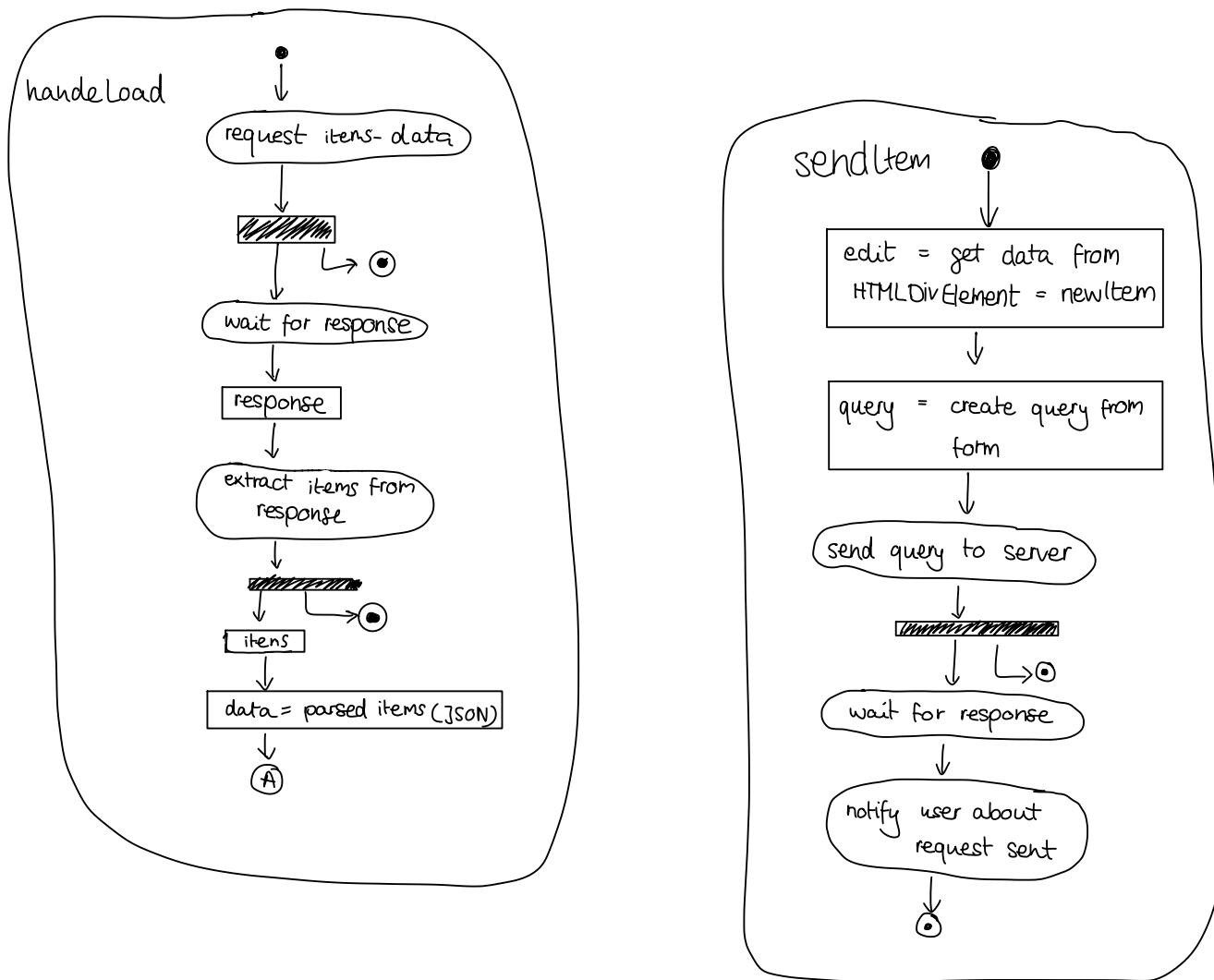
Input
name : string
amount : number
comment : string
date : string



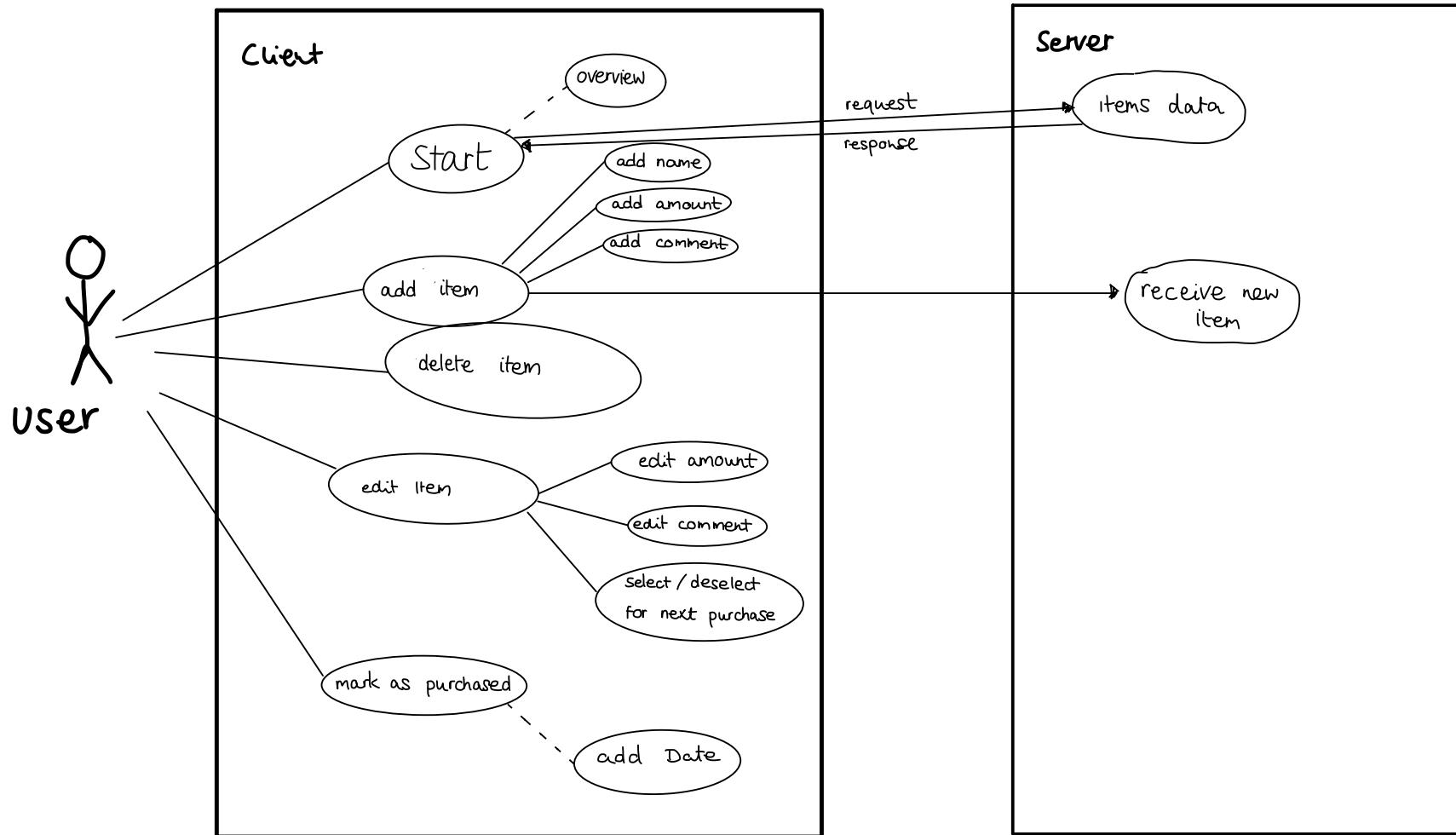
LOS - ShoppingList : Activity Diagramm



LOS - ShoppingList : Activity Diagramm Client



ShoppingList · Use-Case - Diagramm



- PHP hates Programmers
- = Hypertext Preprocessor
- Personal Homepage Tools

ShoppingList · Use-Case - Diagramm

