

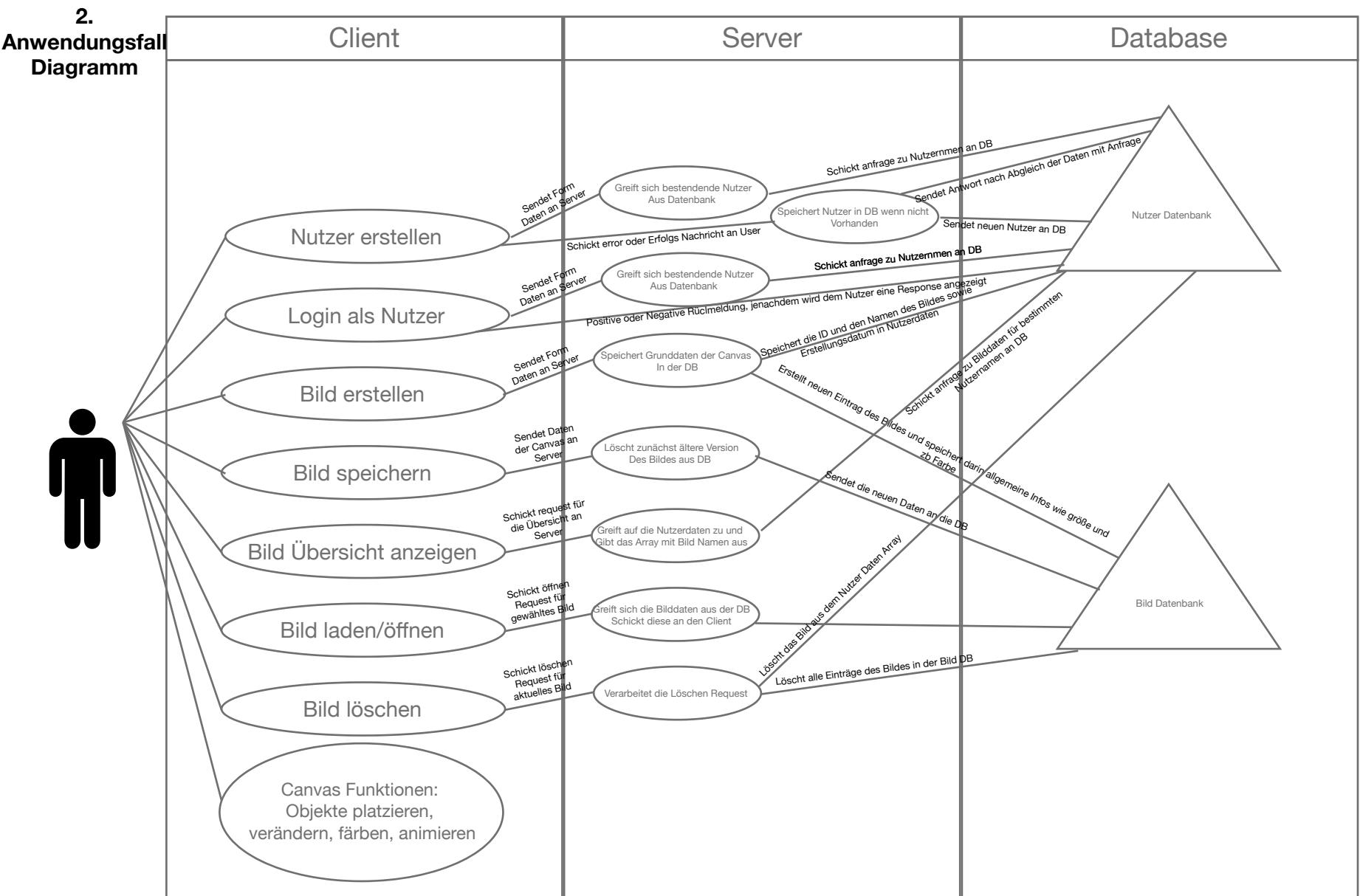
Konzept EIA2 Endabgabe Zauberbild

Inhaltsverzeichnis zu diesem Konzept

1. HTML Skizzen
2. Nutzer/Anwendungsfall Diagramm
3. Schwimmbahnen Diagramme mit zugehörigen Aktivitätsdiagrammen für Kompliziertere Funktionen
4. Notizen
5. Dokumentation

Links zu den externen Dateien:

- Repository auf GitHub:
https://github.com/ivenios/EIA2/tree/master/01_Endabgabe_v2
- Anwendung:
https://ivenios.github.io/EIA2/01_Endabgabe_v2/index.html?
- Konzept Dateien:
- Zip-Datei:
- Dokumentation:
https://github.com/ivenios/EIA2/blob/master/01_Endabgabe_v2/readME.txt

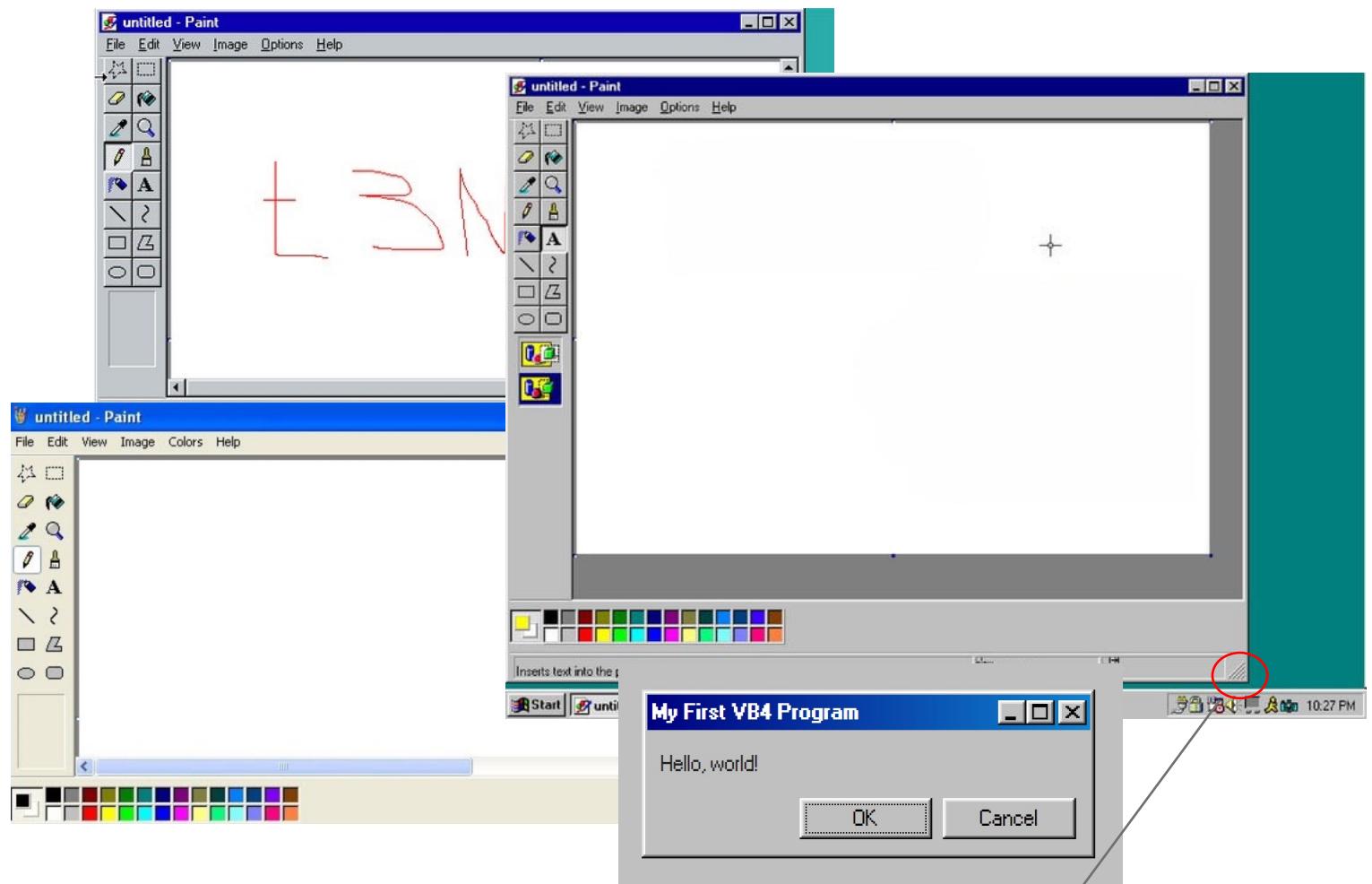


HTML Skizzen

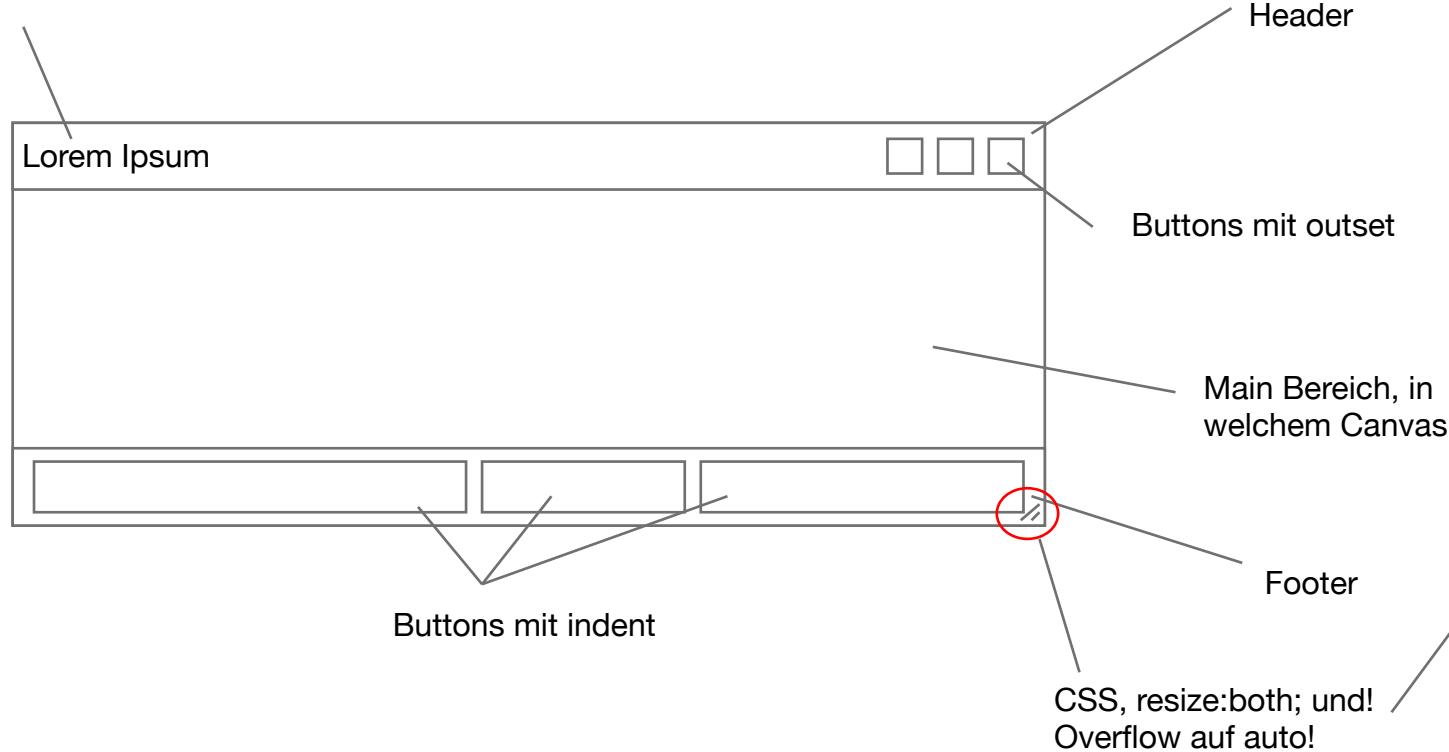
Benötigt wird ein Grundkonzept für den Aufbau für jedes Fenster welches dynamisch aufgebaut wird.

- 1- Willkommens Message mit Erklärung
- 2- Login in Panel
- 3- User Register Panel
- 4- Bilder Übersicht Panel
- 5- Neues Bild erstellen
- 6- Bild Canvas
- 7- Dynamische Error message, in die die passenden Nachrichten eingebaut werden.

Mood Board:



Überschrift



Canvas HTML Skizze:

Change event listener 
Click event listener 
Mousemove event listener 
Mousedown / MouseUp event listener 

Header

Header top

Dynamische Überschrift id=canvasTitle



Header Tab

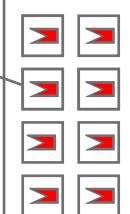
Header Tabs (Blank Buttons)

Section Content

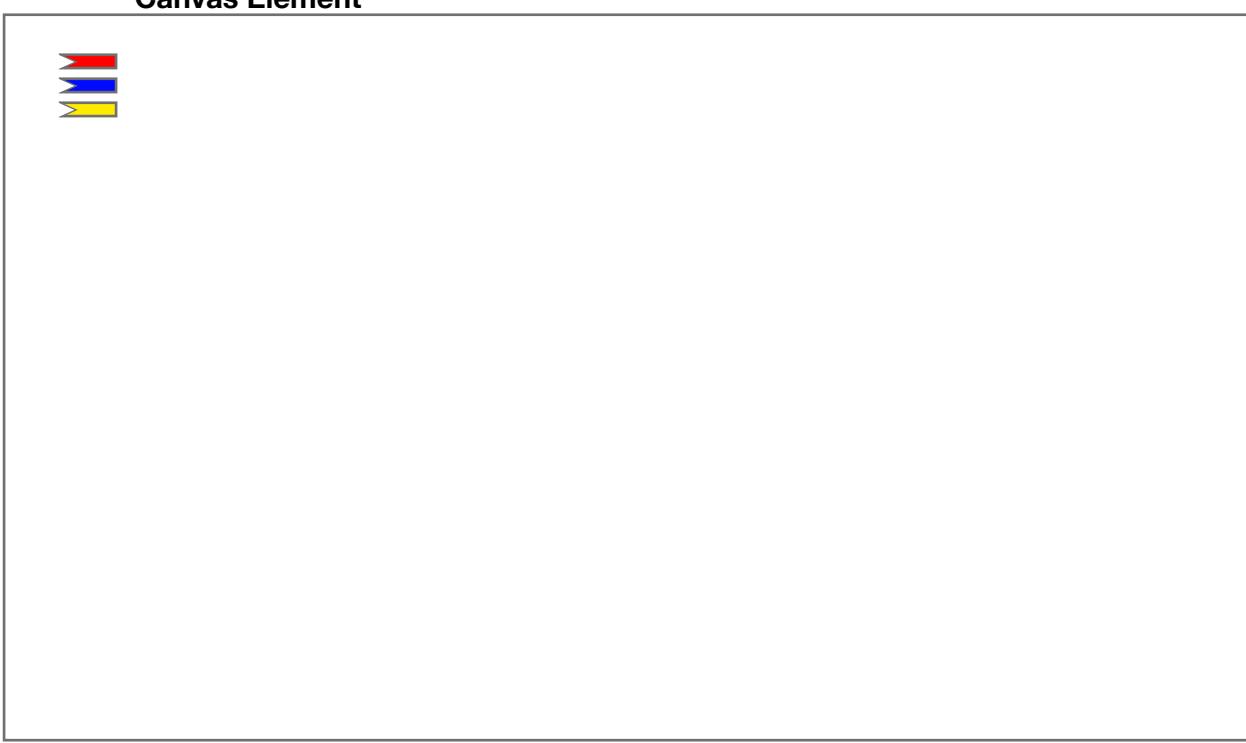
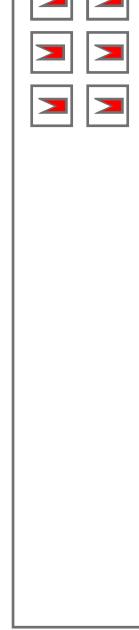
Aside

Canvas Element

Content mittig

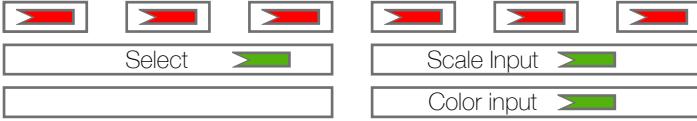


Tools / Tools Bar



Footer

Innen Bereich



Buttons zum speichern animieren und co

Form Elemente für die Farbe, Scale und Animation Typen

Bottom



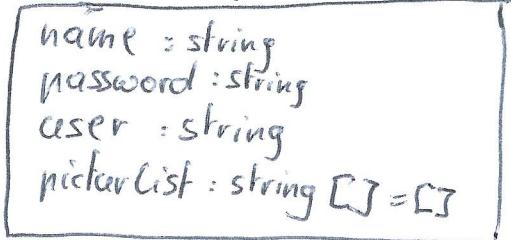
Datenbank:

→ Datenmodelle

⇒ asso. Arrays ~~siehe infotyps.ts~~

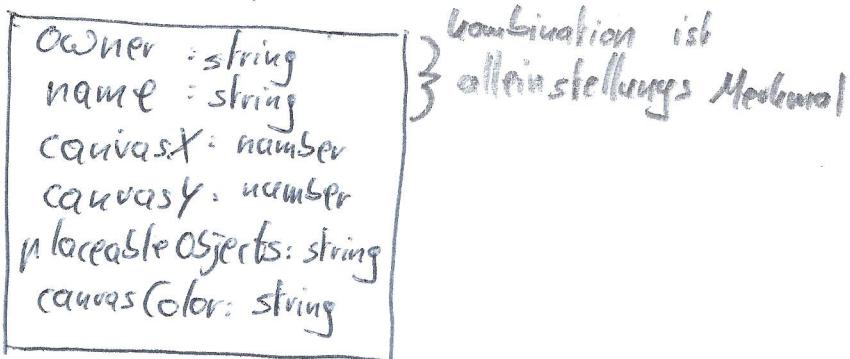
collection Userdatabase

⇒ UserData



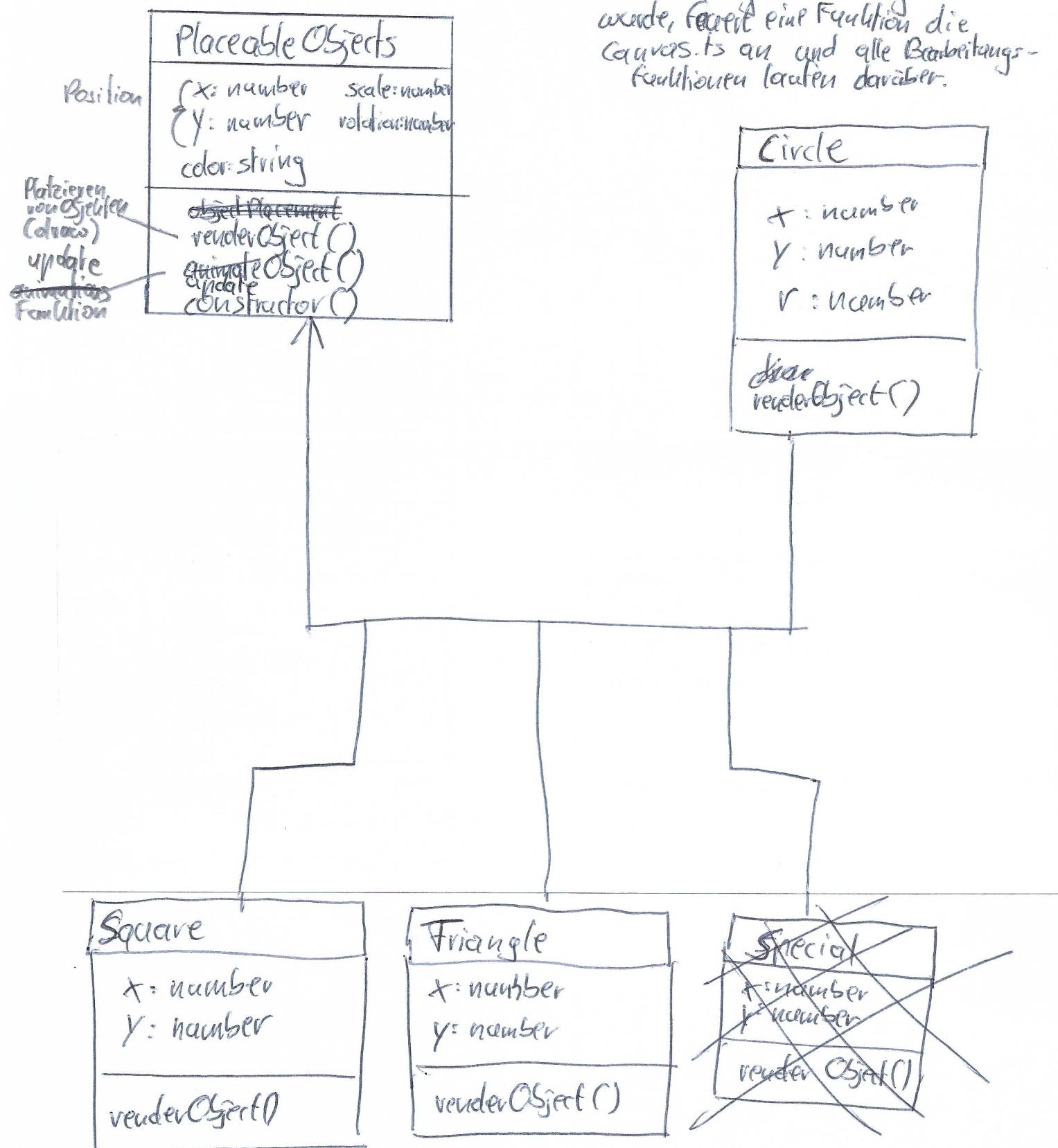
collection canvasDatabase

⇒ Canvas Data



Die Classes von Radle MS Paint

→ canvas.ts dabei auch in zusätzliches File
↳ nach dem die Grandcanvas geladen
wurde, fügt es eine Funktion die
Canvas.ts an und alle Bearbeitungs-
funktionen laufen darüber.

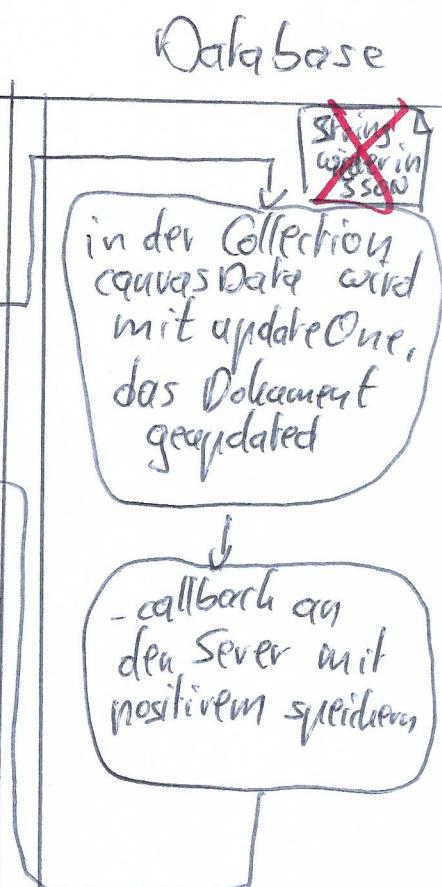
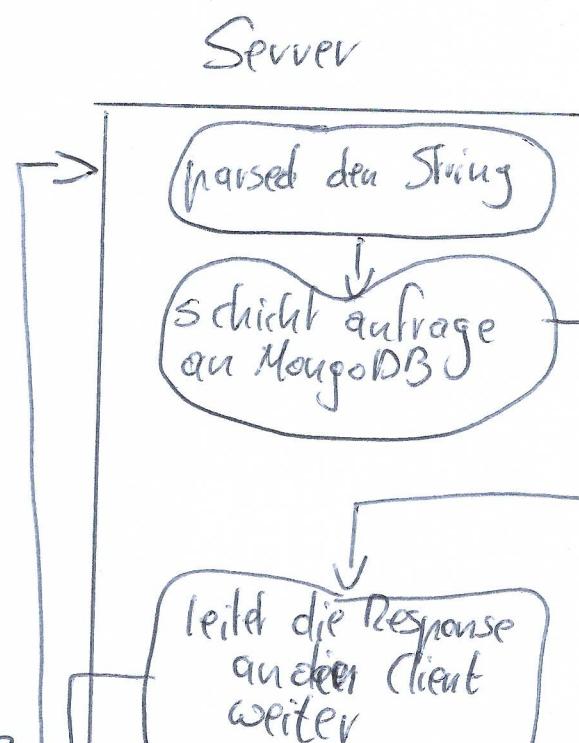
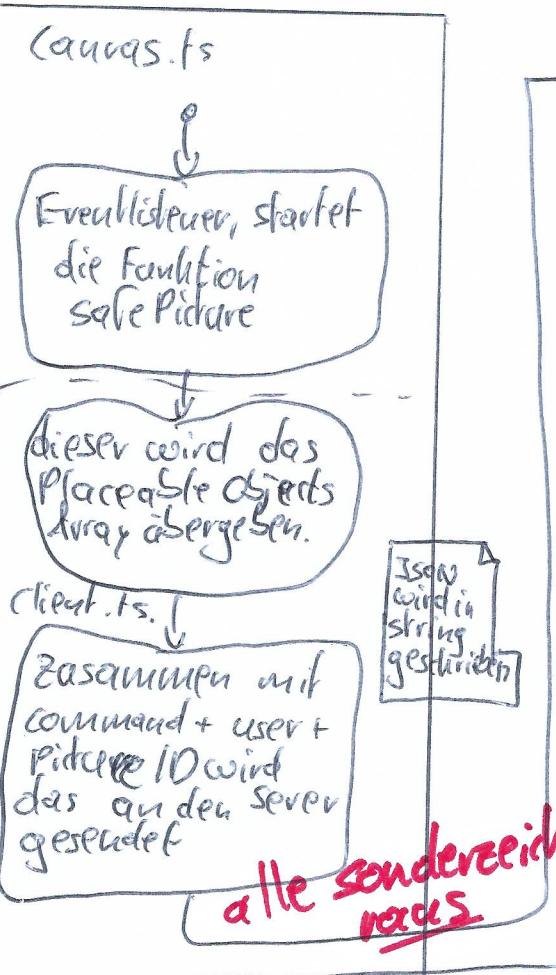


Working Title
→ alle Sachen darstellen

lassen
→ dann Classesstruktur anwenden

Bild speichern

Client



Issue query lange
→ muss man
den Array nach
Veränderung noch
einfach neu gemacht
werden??

→ Wenn nichts
verändert muss
Funktion nicht
ausführen!

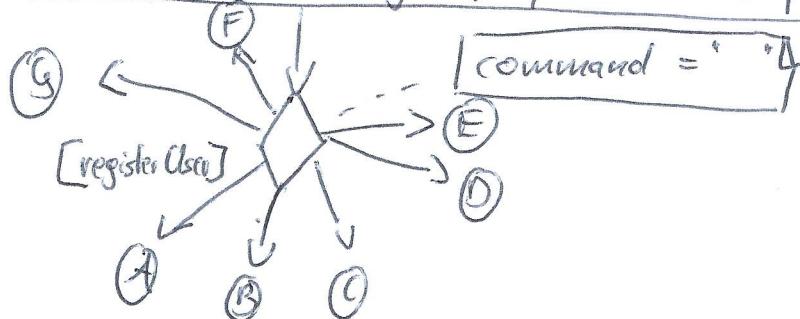
Server seitige Verarbeitung

handlerequest

> Listener request

query Ctrl wird in die einzelnen Schnipsel gepräst

let command = query["command"]



[registerUser]



let NewUser: UserData

queries an die einzelnen Array stellen schreiben

DB.registerUserName(username, findCallback, NewUser)

G [load Selected Picture]

DB.loadPictureFromDBC
findCallback, query["username"], query["pictureName"]

[loginUser]



password: string

DB.loginUser(username, password, findCallback)

[Load PictureList]



DB.loadListFromBB
username, findCallback

[InitiatePicture] [deletePicture], [Safe Picture]



queryString
eigener Nutzernam

CanvasData: CanvasData

DB.deletePictureFromDB
username, pictureName, findCallback

-user, -pic,
query["placeableObject"]

DB.safePictureToDBC
findCallback,

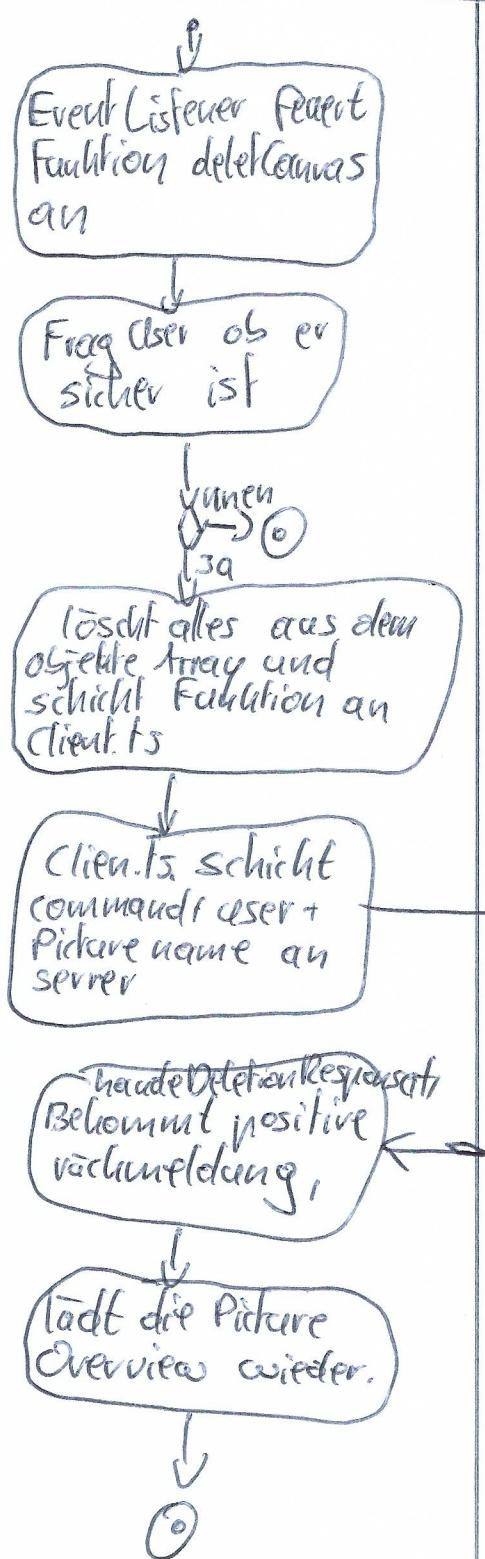
queries in das Array schreiben

DB.pushPictureToDBC
CanvasData, findCallback

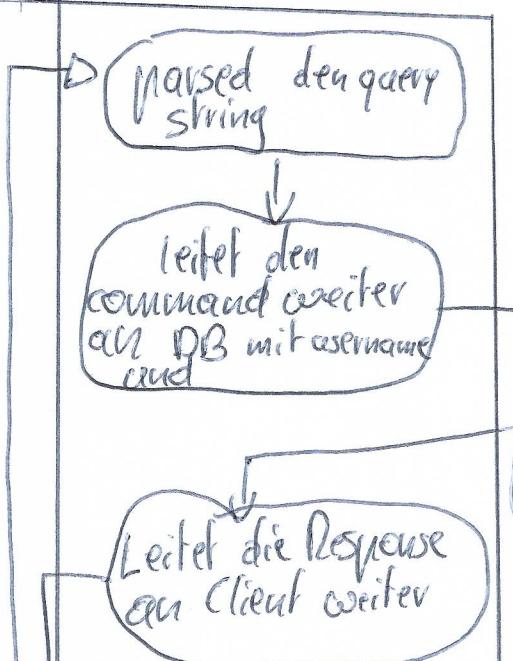


Bild löschen

Client



Server



Database

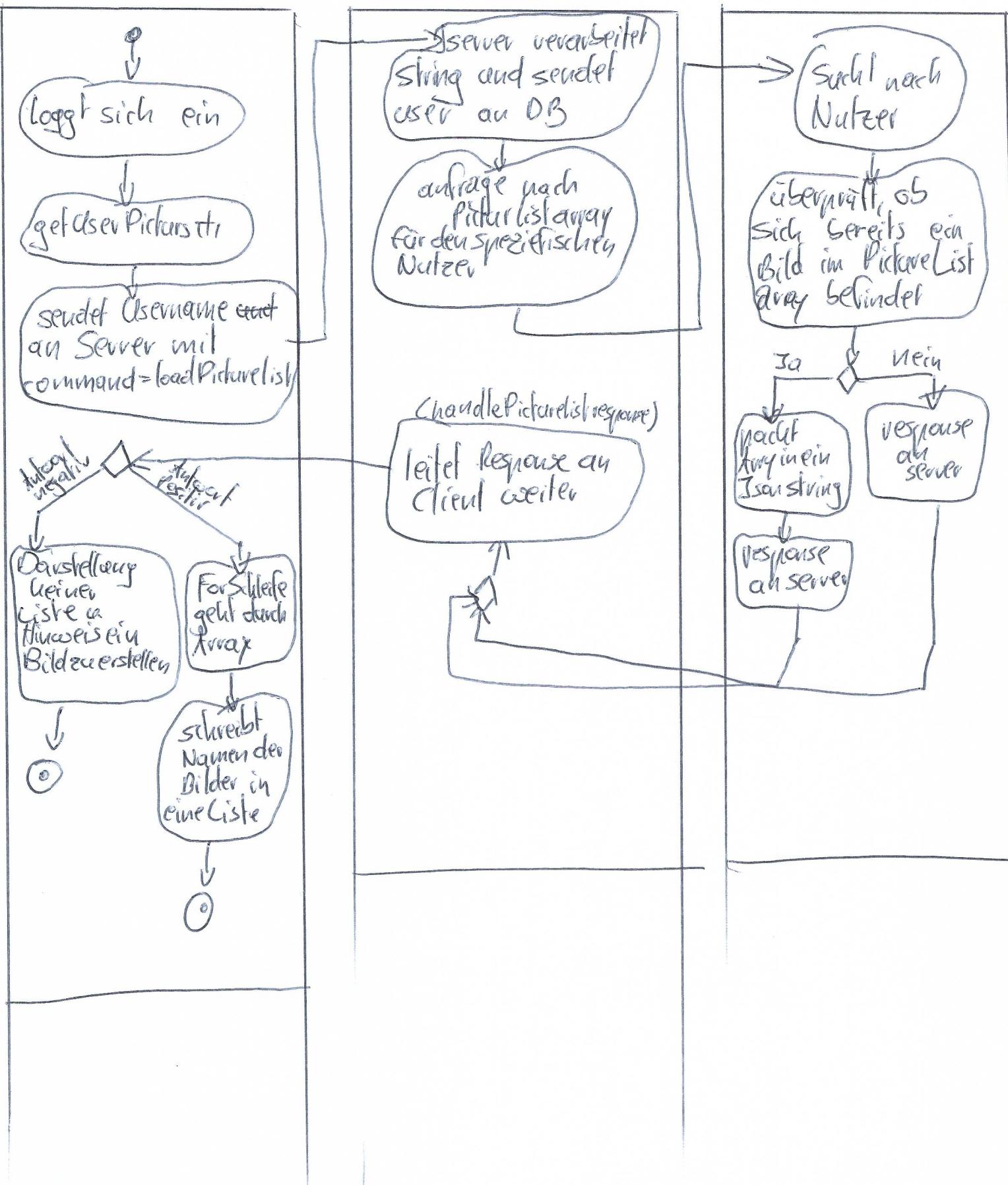


Bild übersichts Liste laden

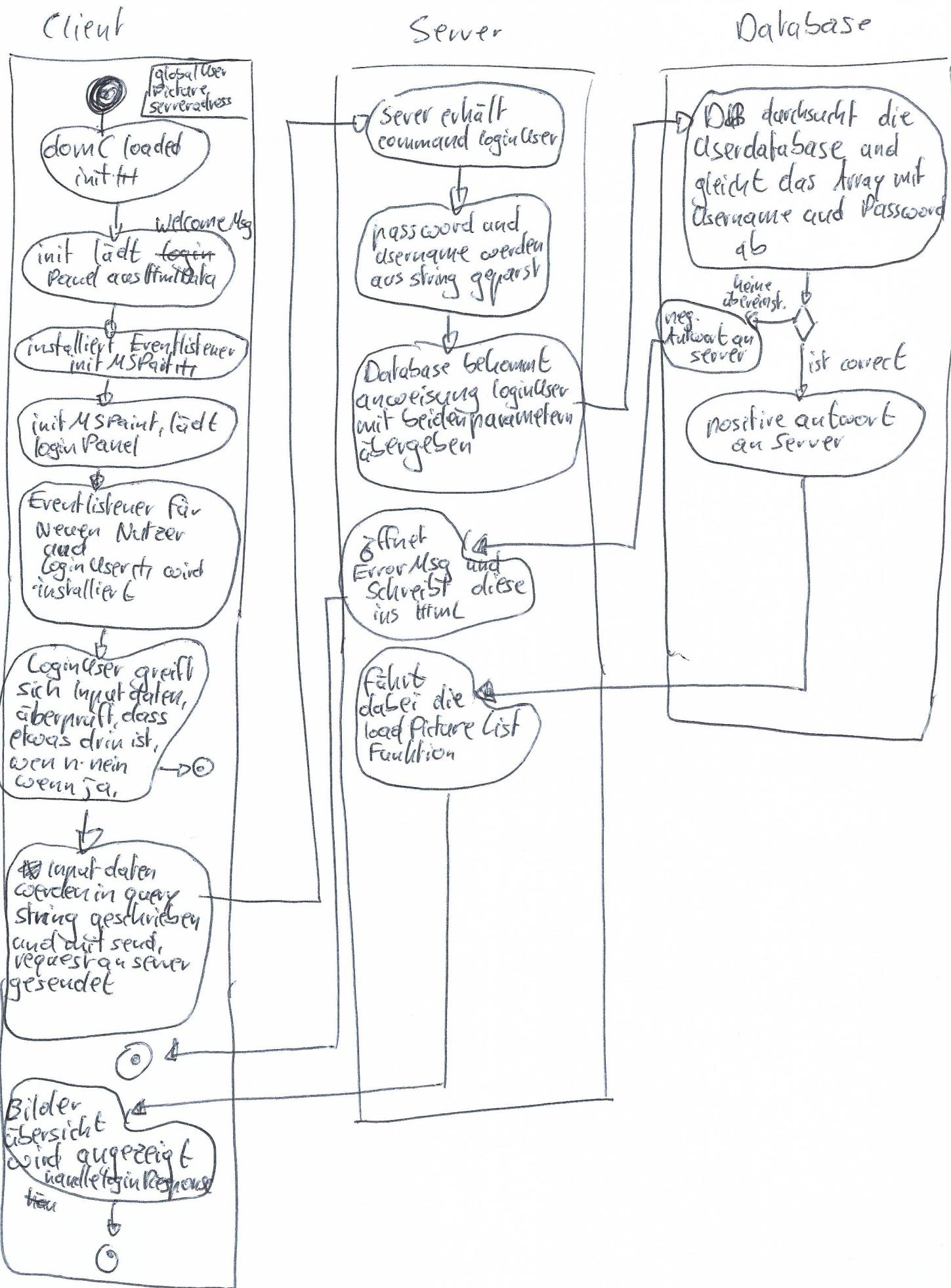
Client

Server

Database

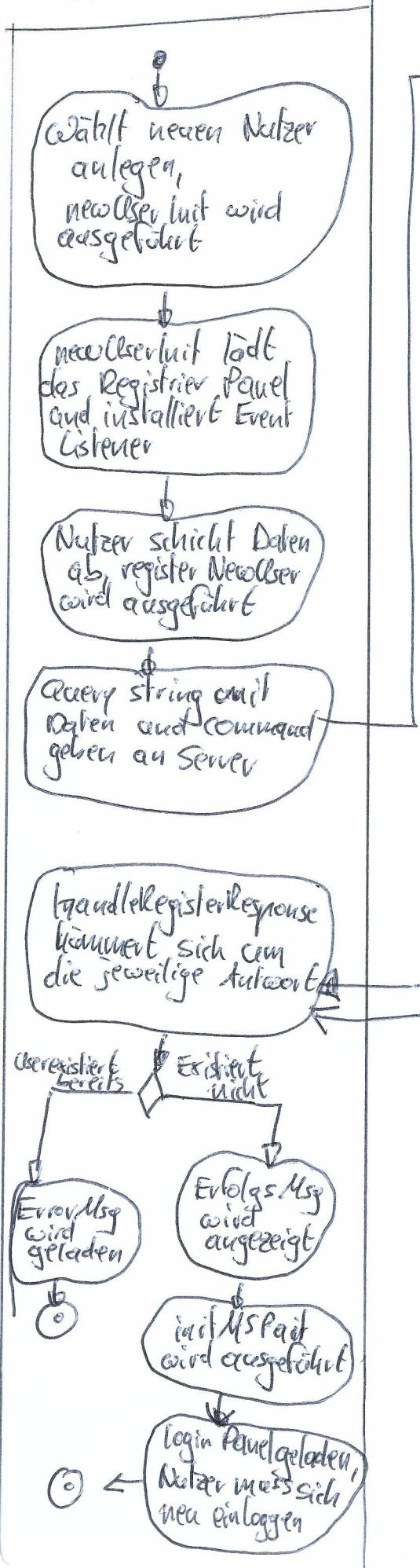


Login Funktion 1 Start

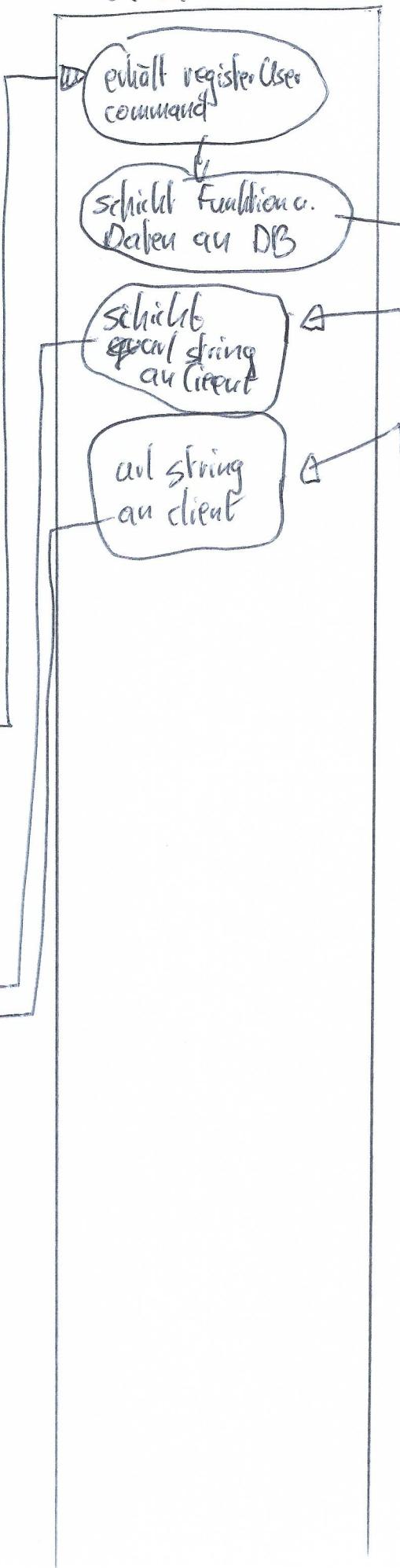


Neuen Nutzer registrieren

Client



Server



DB

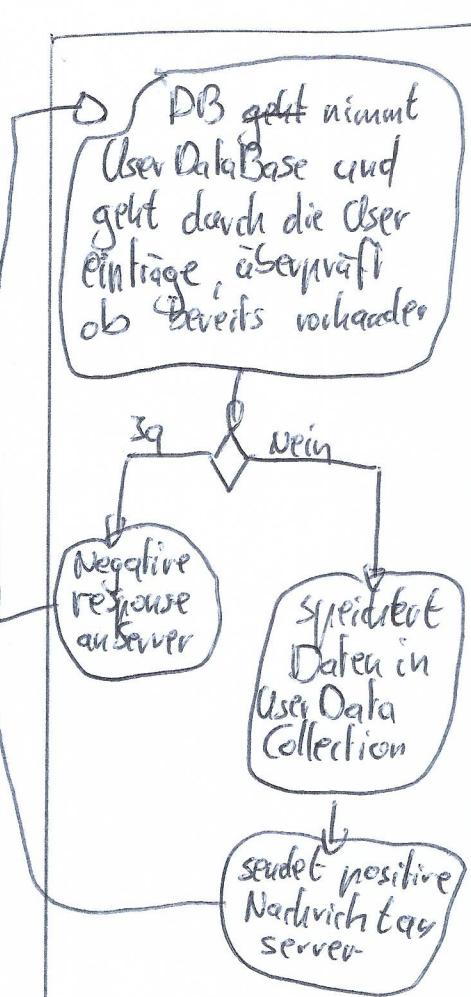


Bild anlegen / Canvas erstellen

Client

Server

Datenbank

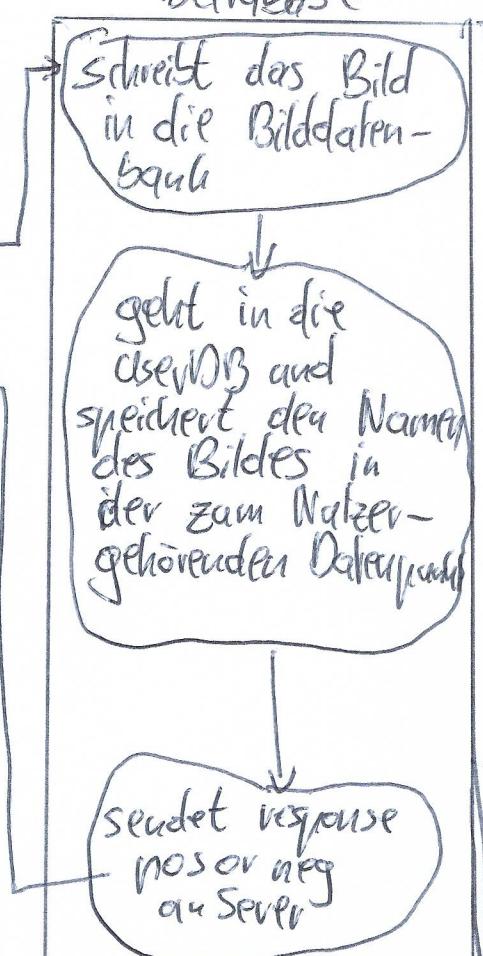
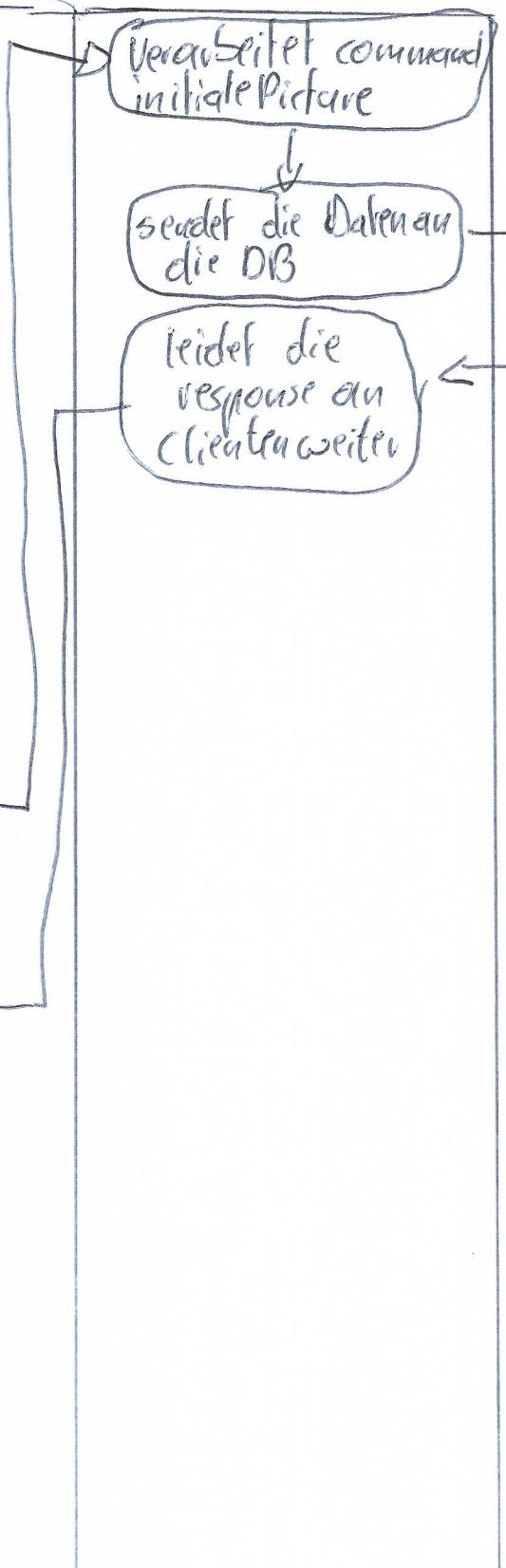
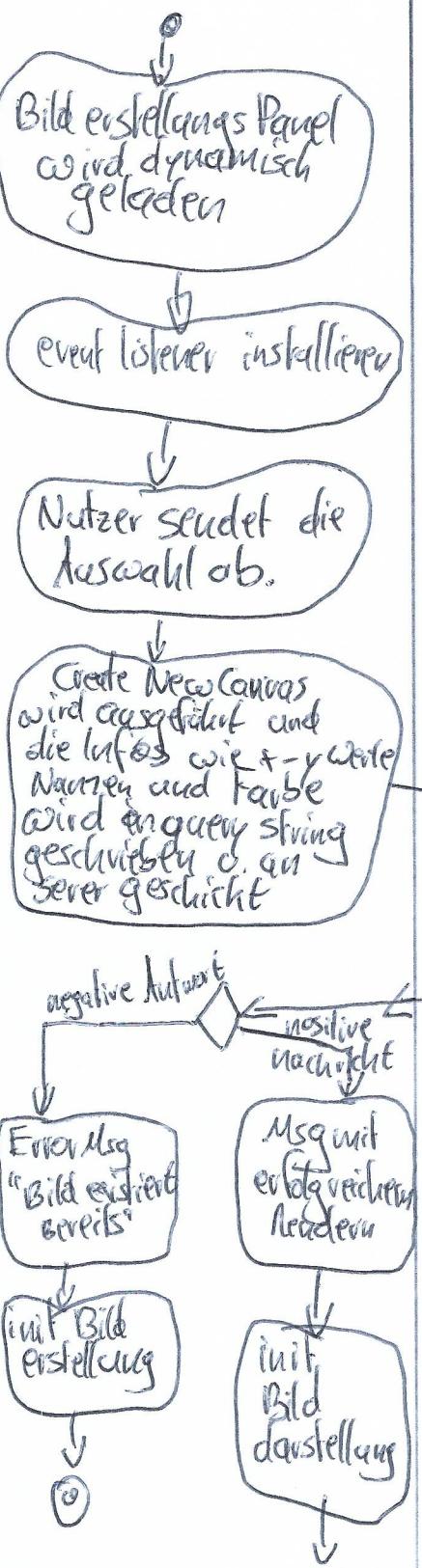


Bild & das erstmal/ erstellen und laden:

global variables

> click: createNewCanvas

```
let globalUser: string  
let globalPictureName: string  
let canvasSizeX: number  
let canvasSizeY: number  
let canvasColor: string
```

```
let re: string = "#"  
let query: string = "command=initiatePicture"  
let inputs: HTMLCollection
```

```
let queryColor: string = inputs[3].value
```

```
queryColor = queryColor.replace(re, "%23")
```

restliche inputs mit den gewöhnlichen
query keys in der query schreiben
und die globalen Variablen

```
query += `&${"canvasColor"}=${queryColor}
```

Send Request Query, handle NewCanvas Response

New
handle Canvas Response

Wenn der Server "Save Positive" weitergeleitet
hat, wird initRenderCanvas geführt

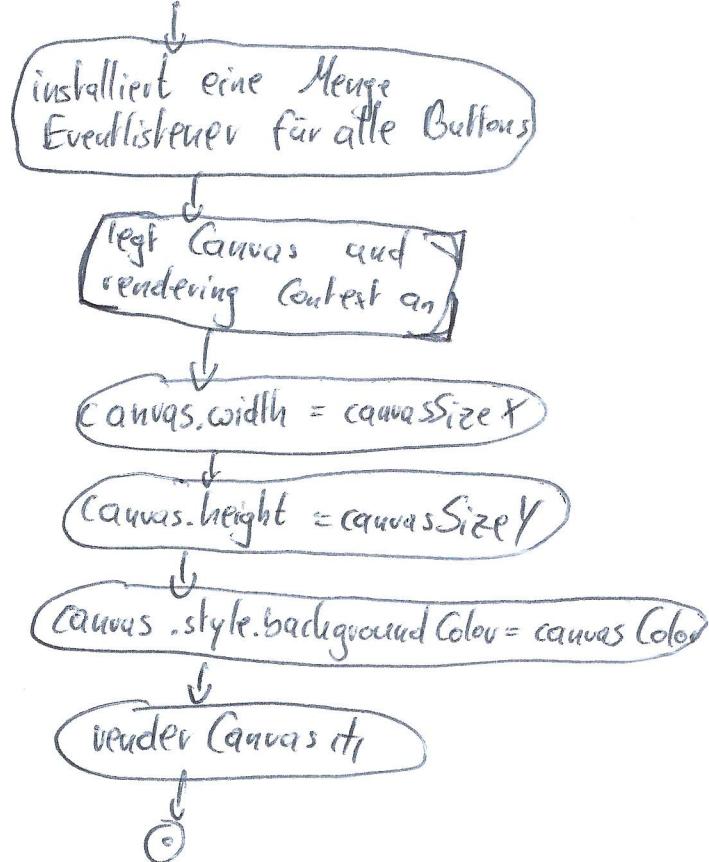
initRenderNewCanvas

innerHTML wird ersetzt durch htmlData["mainCanvasPanel"]

initCanvas()

in der
canvas.js

init Canvas



delete Canvas

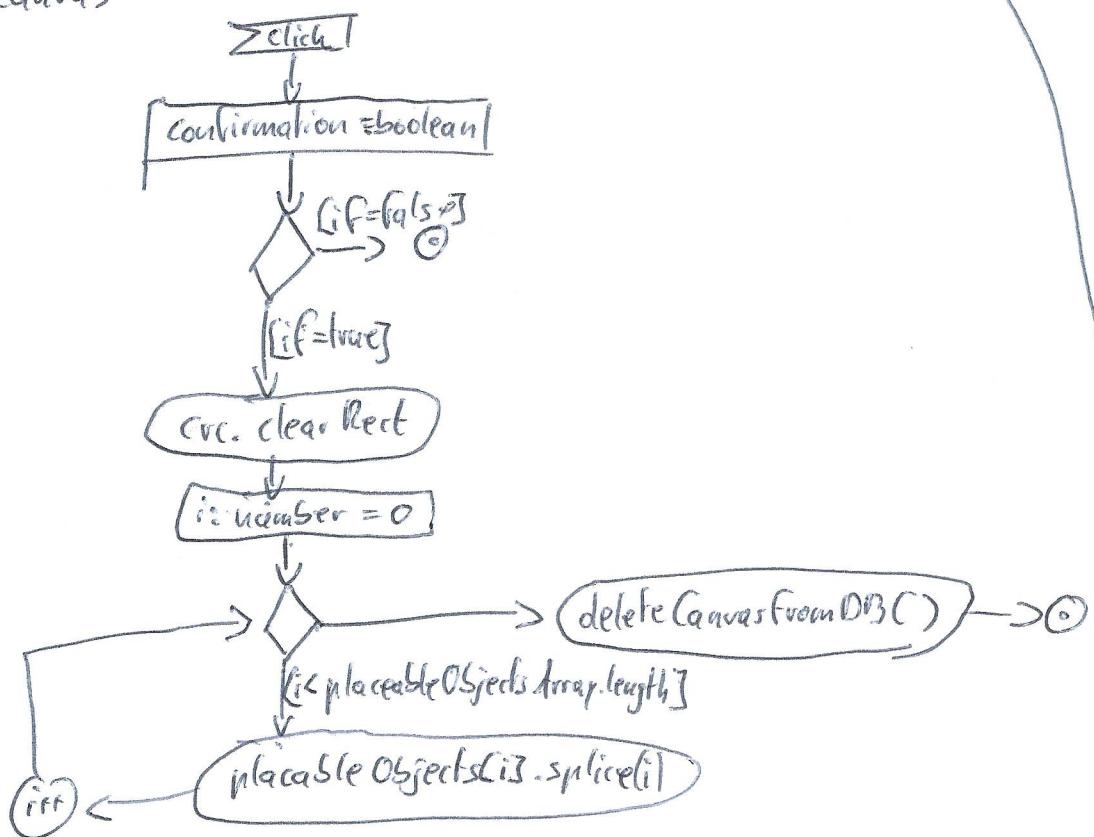
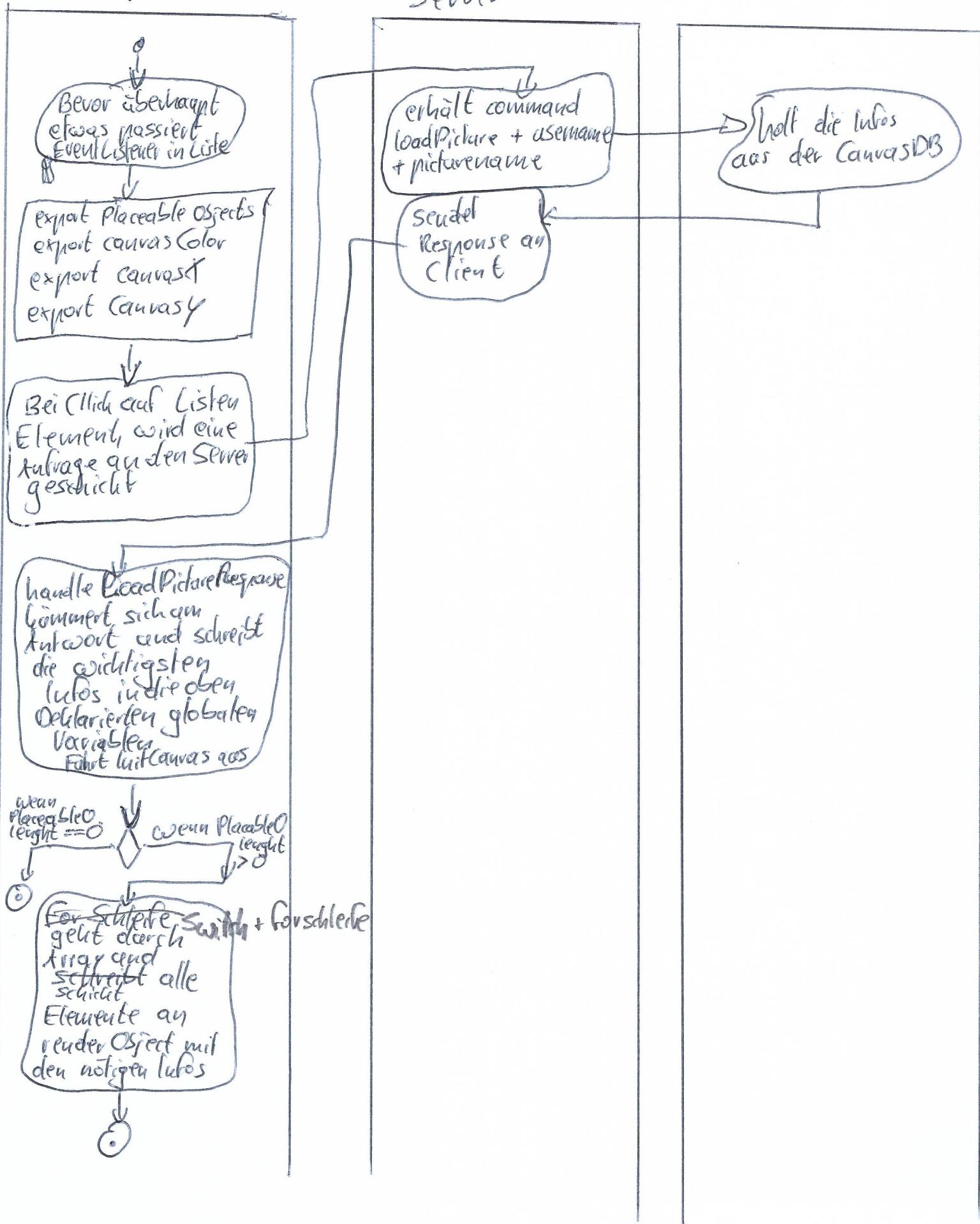


Bild erneut bearbeiten:
⇒ woher weiß der Button, welche ID gedrückt wurde?

Client

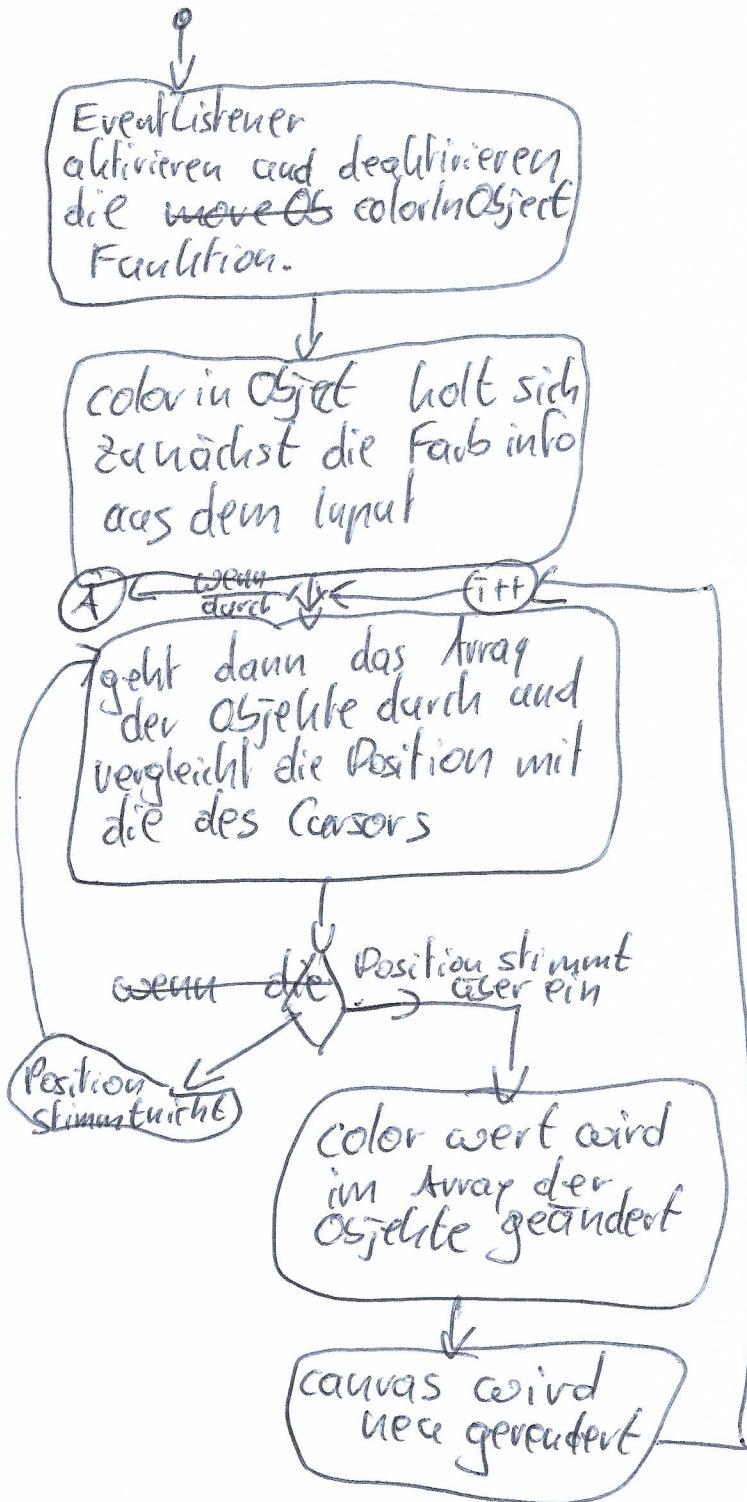
Server

Database



Objekte Färben:

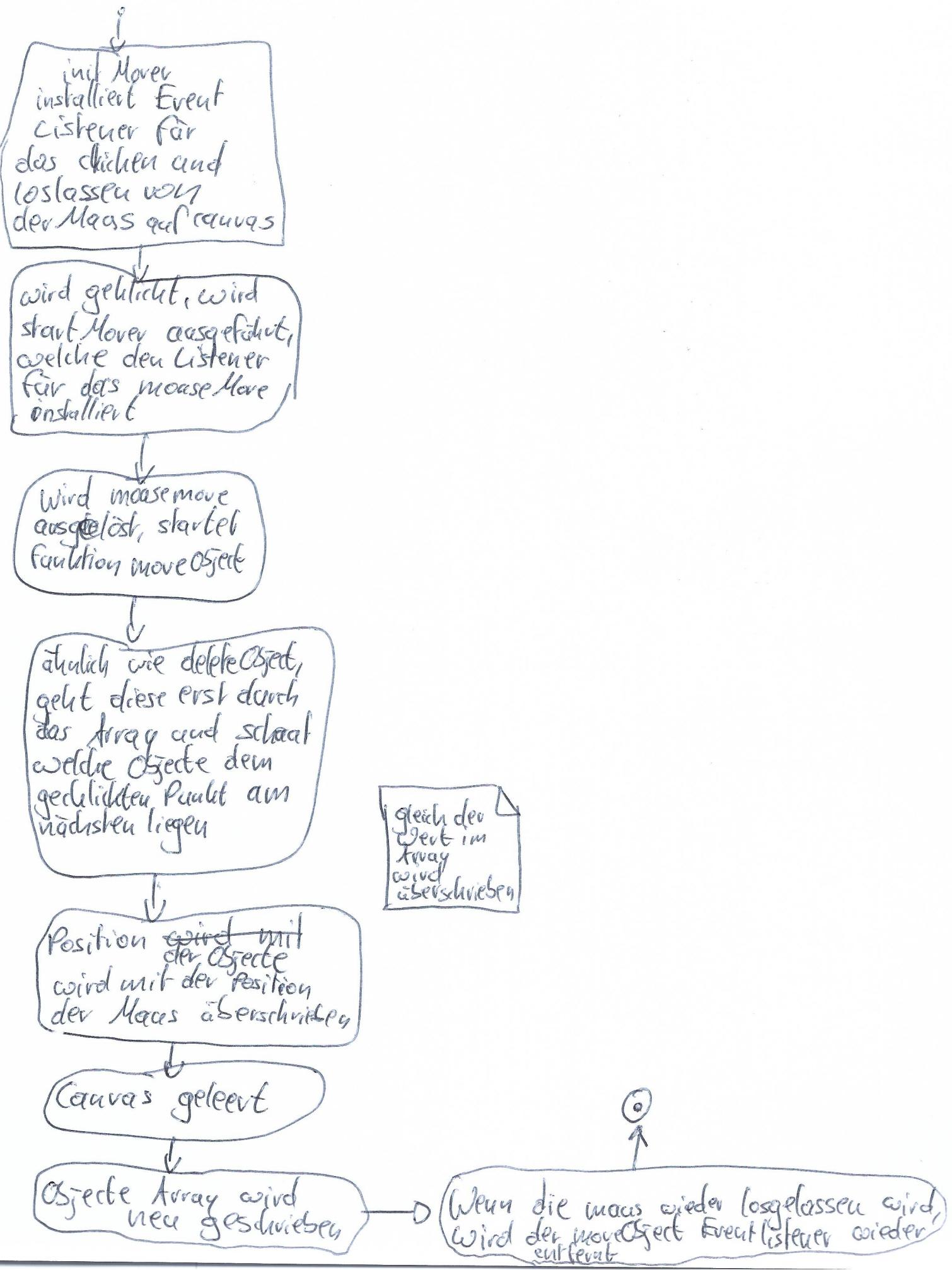
⇒ im großen und ganzen die Bewegen Funktion



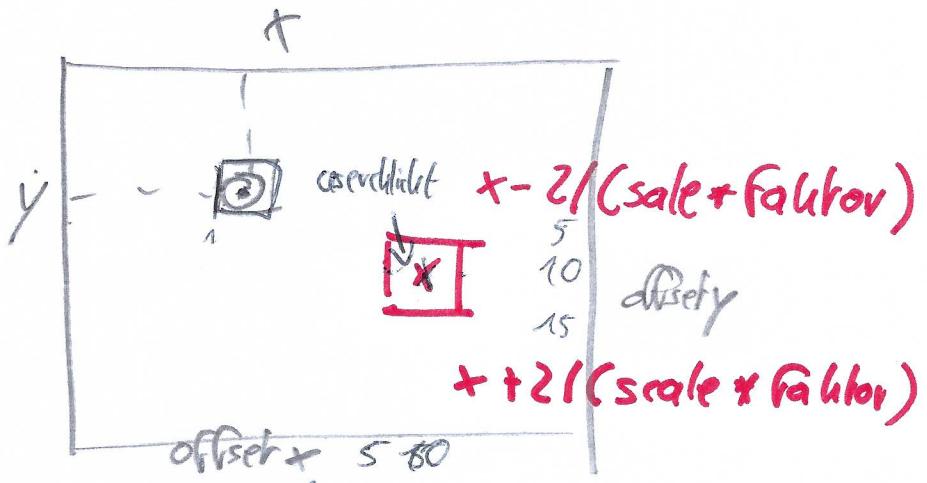
Änderung des Array
⇒ selber Anatz,
nur wird die
type Variable im
Objekte Array um
geschrieben.

Bewegen von Objekten

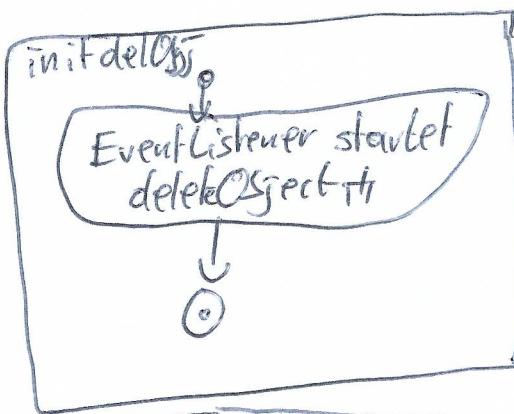
→ Grund daran kommt vom Löschen



Objekt von Canvas löschen.

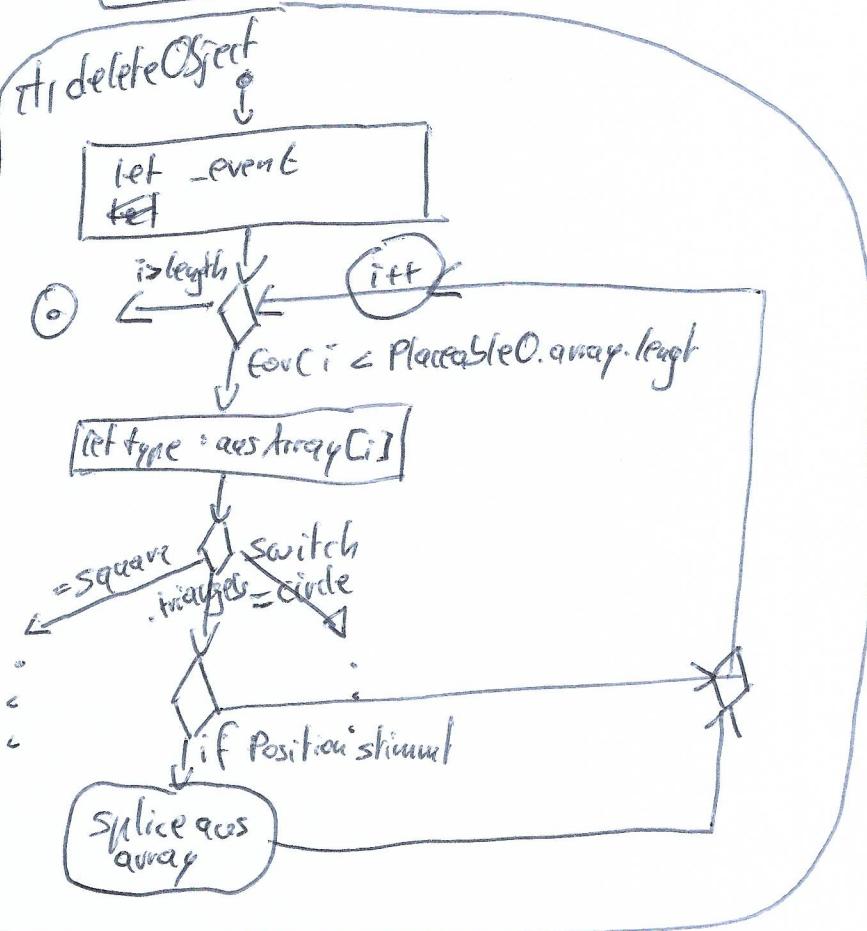


Für x : $\frac{\text{array}.x - \text{scale} * \text{faktor}}{\text{array}.x + \text{scale} * \text{faktor}} \geq \text{offset}$ \wedge
 $\text{array}.x + \text{scale} * \text{faktor} \leq \text{offset}$
⋮



→ in if clauses berechnet es komischerweise falsch
⇒ Lösung zu nächst & in Variablen berechnen

iFsizeX
iFsizeYm
iFsizeYf
iFsizeYm



z.B.

X 54 088

assert = 55

+ 62

X 53

Bild animieren lassen

Bild Animation pausieren

canvas.ts

canvas.ts

Event-Listener für Motor Animer
Button

Funktion updateObject()
Fügt feuer ~~an~~ e in einer
Forschleife die einzelnen
Objekt updatet für
jedes Objekt im
Object Array

abfrage wo sich der
Animation Counter befindet

ist 1
Stop Animation

ist 2
Start Animation

geht mit
window.setTimeout(handler, time, true/false)

→ das Bild muss
nach Bewegung
neu gerendert
werden
dafür muss
das Objekt Array
aktualisiert werden.

First draft

1. HTML-Skizzen

5. Dokumentation

*****RUDE MS PAINT*****

- - - - -

by Iven Otis Sieglen.

This programm includes:

1. Konzept
2. Dokumentation
3. zip-Datei mit Abgabe
4. Das Programm

This is the

~~~~~Documentation~~~~~

Please visit the link down below,  
to get to the other documents

[https://github.com/ivenios/EIA2/tree/master/01\\_Endabgabe\\_v2](https://github.com/ivenios/EIA2/tree/master/01_Endabgabe_v2)

- - - - 1. How to Play - - - -

If you want to start right away,  
just open the link down below:  
[https://ivenios.github.io/EIA2/01\\_Endabgabe\\_v2/index.html?](https://ivenios.github.io/EIA2/01_Endabgabe_v2/index.html?)  
If you are a special person, you  
can also install it on your own  
heroku and mongoDB services.  
For that skip to 2.

\*Important Notice\*

When you are asked to log in or to  
create a new user, never ever ever  
use a real password. Use something

like "123" or "uuuu". Because this programm sends server Requests via GET with an query string, everyone who wants to maybe harm you in some way, can get your password with ease.

#### \*Step by Step Guide\*

You can controll the programm with just your mouse (+ your keyboard for typing).

1. You are greeted with a welcome message

2. You will be asked to log in or with the "Im new here" button, you can create a new user.

3. Your very own picture overview will load.

If your new, there will be nothing, but you can create your first pictrue with in seconds. If you already have created some picture, the names of the pictures will be loaded.

4. Creting a picture. When you choose to create a new picture, a panel with some customizations will be loaded, where you can choose the background color and the x and y size of your canvas in pixels.

5. After you clicked "Create new canvas", you will be greeted with your picture.

#### \*The Controls\*

Here you can see a represantation of the tool button placement aside the canvas

[1][2]

[3][4]

[5][6]

[7][8]

[9][0]

- 1 - Place a square anywhere
- 2 - Place a circel anywhere
- 3 - Place a triangel anywhere
- 4 - The mover. With drap and drop move already existing objects
- 5 - The rubber. Will remove objects when you click on them
- 6 - Empty
- 7 - The spray can. Change the color of objects that already placed
- 8 - The resizer. Resize already existing objects by clicking.
- 9 - Empty
- 0 - Epmty

Tip: With the color input and the scale slider,  
you can alter the color and scale of objects that you are  
about to place. Or with the appropriate tools from above,  
you can change the color and scale afterwards.

#### \*Animation\*

If it works, you should be able to let the canvas,  
"do its thing" and move around the placed objects.  
With the drop down selector, you can choose between  
different animation patterns. Try them out!

#### \*Saving and deleting\*

Make sure to always end the animation before saving.  
Always save before quitting!  
A deleted picture can't be brought back!

## -\_-\_-2. How to Install-\_-\_-

#### \*Things you'll need\*

- a gitHub repository
- a heroku app which is connected with

the above mentioned repository

- a cluster with some space on mongoDB
- an installed compiler, which complies the TypeScript you are about to change

**\*Step by Step\***

1-take the zip-File and unpack it, then put it  
in your new repository

2-in the folder 2\_Program > JS, you will find  
all necessary ts files

3-go to the "client.ts" line 16 and change the  
"serverAddress" to your heroku app link

4-go to the "package.json" and make sure, the  
"server.js" in the js folder is still  
correctly linked

5-now go to the "database.ts" file and change

Line 19 databaseURL - you should get this from mongoDB in your cluster at "connect" in  
the cluster overview

but you first have to create a user at "Database access"

Line 20 databaseName - put in the name of your database inside the Cluster

6-create two collections with the names:

Userdatabase

canvasDatabase

7-push all changes to GitHub

8-deploy your heroku app

You should be ready to go