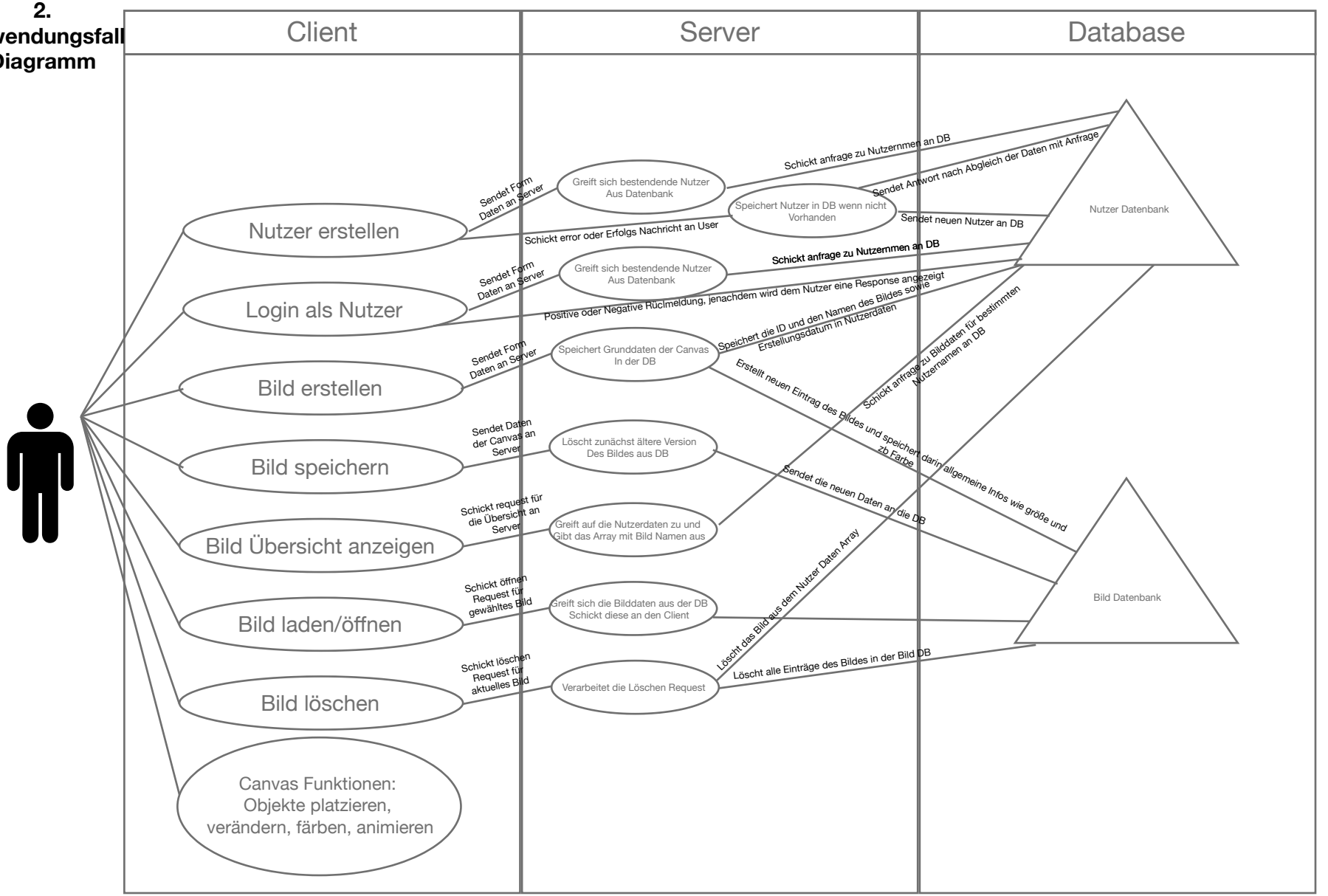
Konzept EIA2 Endabgabe Zauberbild

Inhaltsverzeichnis zu diesem Konzept

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: https://github.com/ivenios/EIA2/tree/master/01_Endabgabe_v2/1_Konzept
- Zip-Datei: https://github.com/ivenios/EIA2/blob/master/01_Endabgabe_v2/Endabgabe%20v2.zip
- Dokumentation: https://github.com/ivenios/EIA2/blob/master/01_Endabgabe_v2/readME.txt

2. Anwendungsfall Diagramm

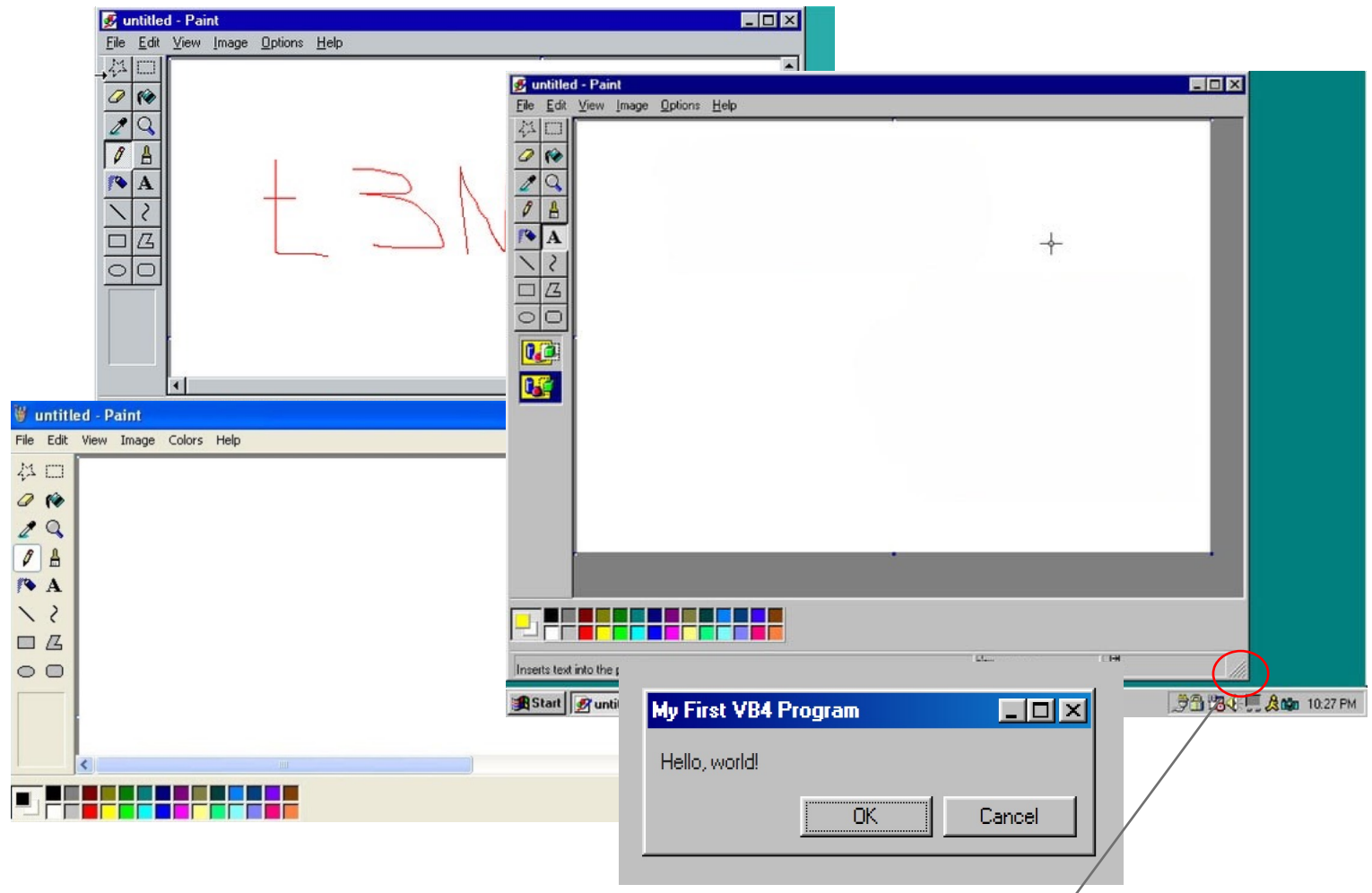


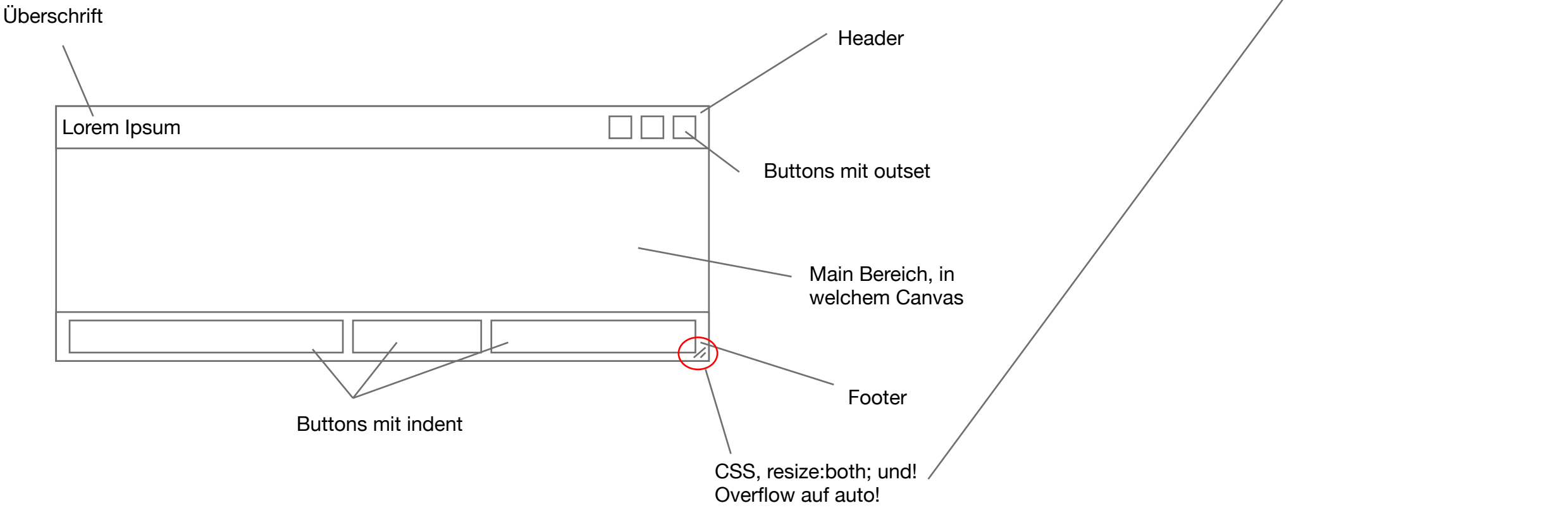
HTML Skizzen

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

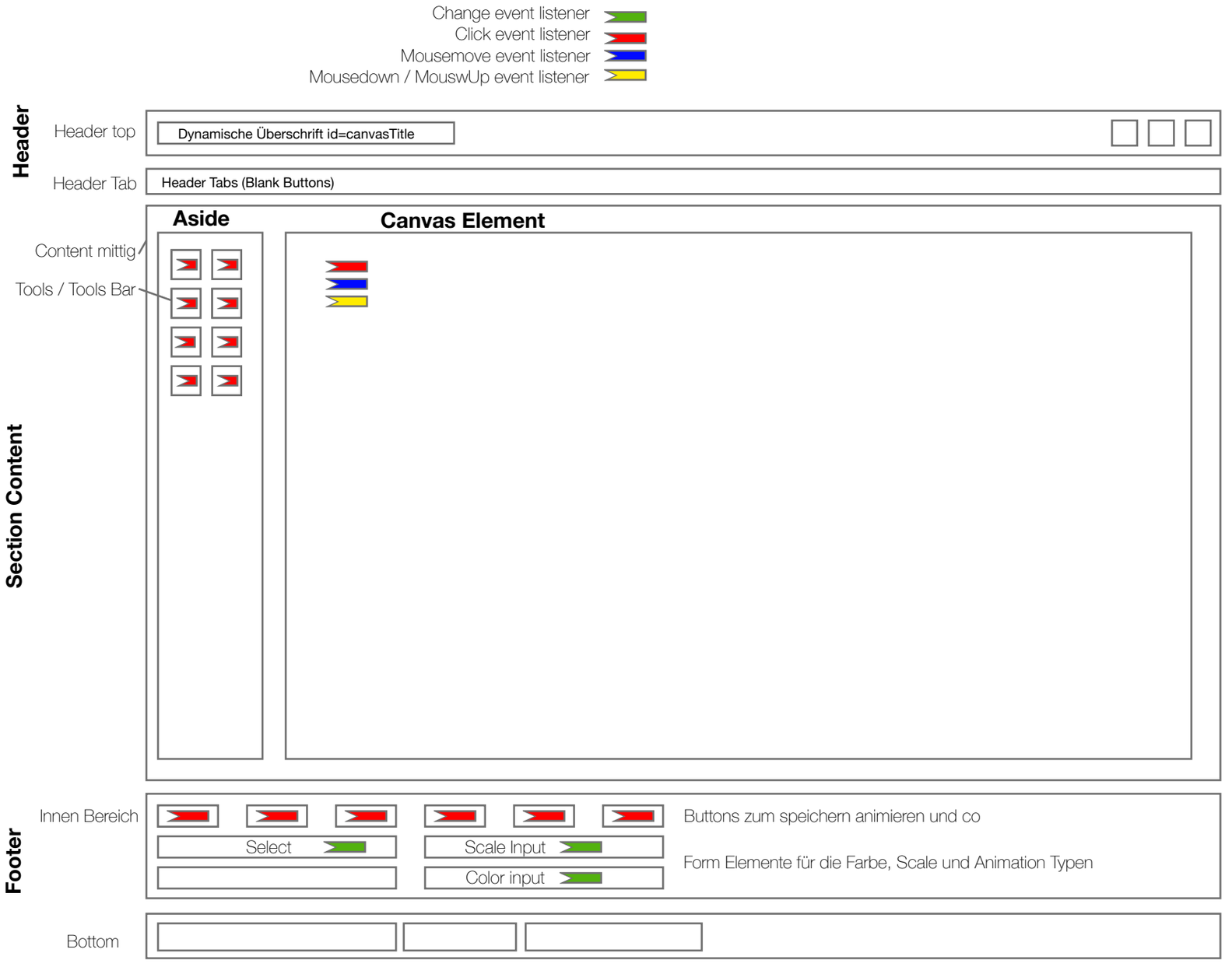
1- Willkommens Message mit Erklärung
2- Login in Panel
3- User Register Panel
4- Bilder Übersicht Panel
5- Neues Bild erstellen
6- Bild Canvas
7- Dynamische Error message, in die die passenden Nachrichten eingebaut werden.

Mood Board:





Canvas HTML Skizze:



Datenbank:

-> Datenmodelle

=> asso. Arrays ~~sichte~~ in typs.ts

collection Userdatabase

=> UserData

```
name : string
password : string
user : string
pictureList : string [] = []
```

collection canvasDatabase

=> CanvasData

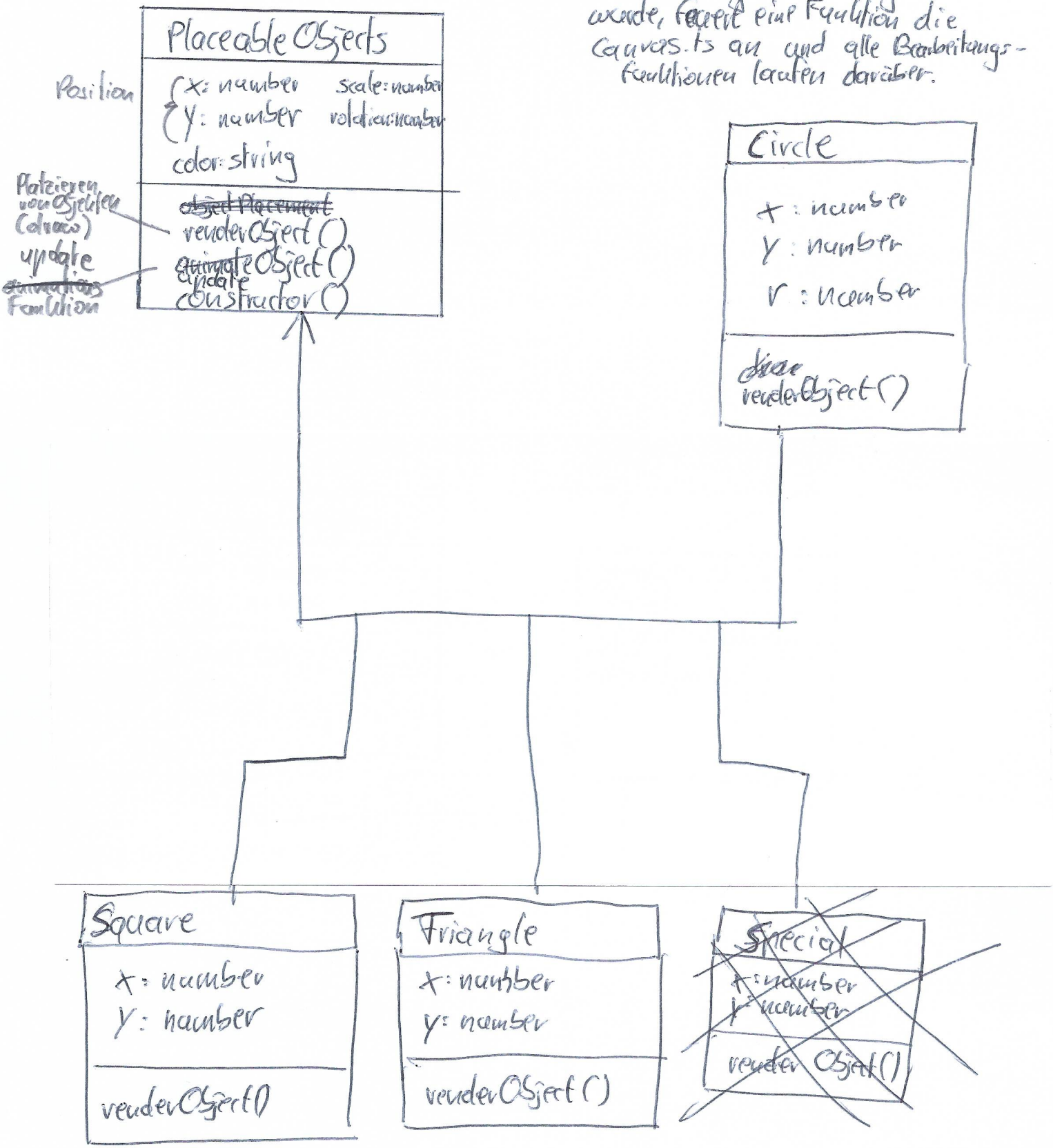
```
owner : string
name : string
canvasX : number
canvasY : number
placeableObjects: string
canvasColor: string
```

} Kombination ist alleinstellungs Merkmal

Die Classes von Rede MS Paint

↳ canvas.ts dabei auch in zusätzliches File

↳ nach dem die Grundcanvas geladen wurde, feuert eine Funktion die canvas.ts an und alle Bearbeitungs-Funktionen laufen darüber.



Working Title

↳ alle Sachen darstellen lassen

↳ dann Classesstruktur anwenden

Bild speichern

Client

Server

Database

canvas.ts

Eventlisteuer, startet die Funktion savePicture

dieser wird das placeable objects Array übergeben.

client.ts.

Zusammen mit command + user + pictureID wird das an den Sever gesendet

Json wird in string geschrieben

alle sondezeichen zwar ersetzen mit %xx?

schreibt dem Nutzer, dass das Speichern erfolgreich war.

parsed den String

schickt anfrage an MongoDB

leitet die Response an den Client weiter

in der Collection canvasData wird mit updateOne, das Dokument geupdated

callback an den Sever mit positivem speichern

-Dsse query länge

=> muss das Array nach veränderung noch einmal neu gemacht werden??

=> Wenn nichts verändert muss Funktion nicht ausführen!

Server seitige Verarbeitung

handleRequest

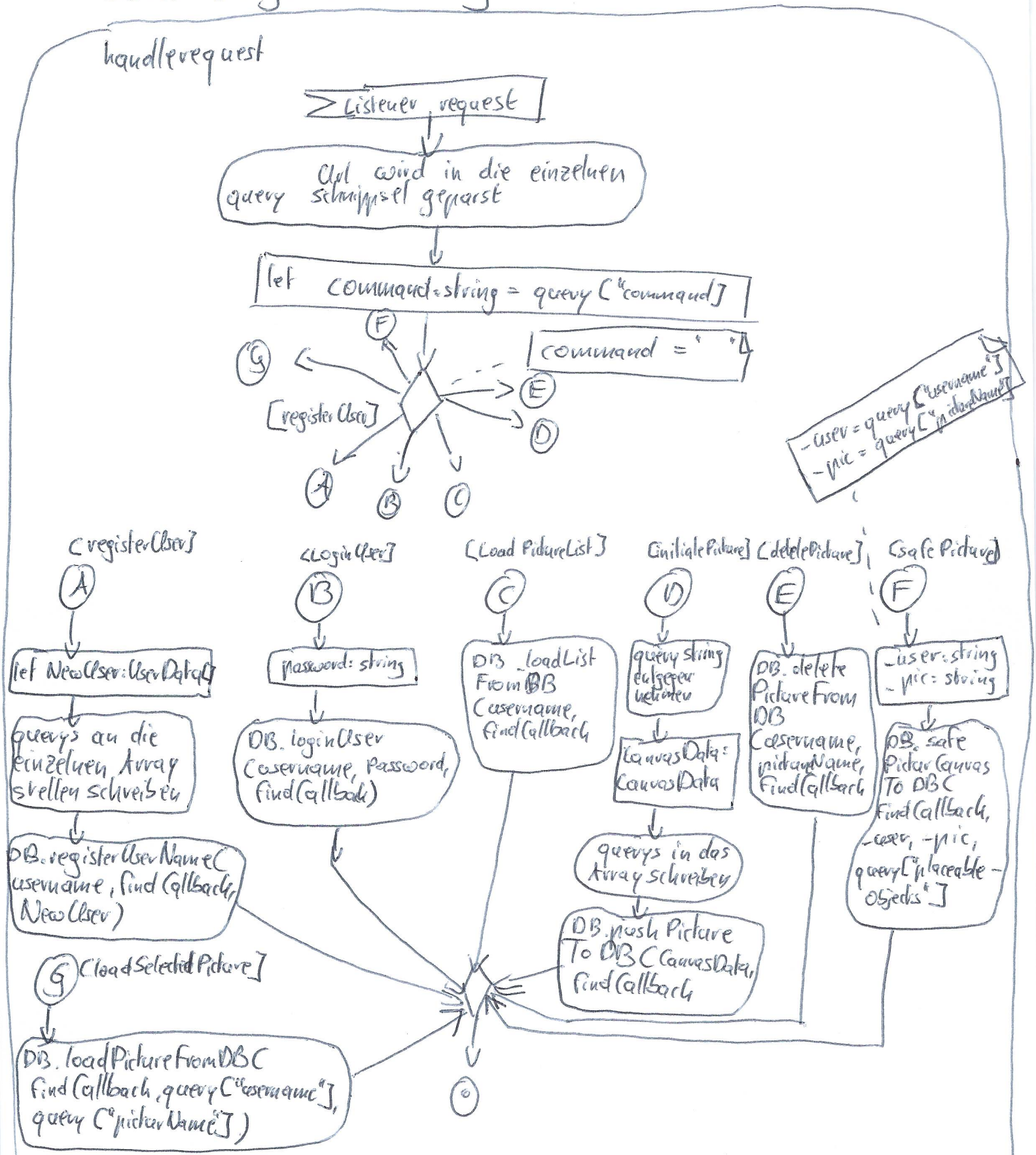


Bild löschen

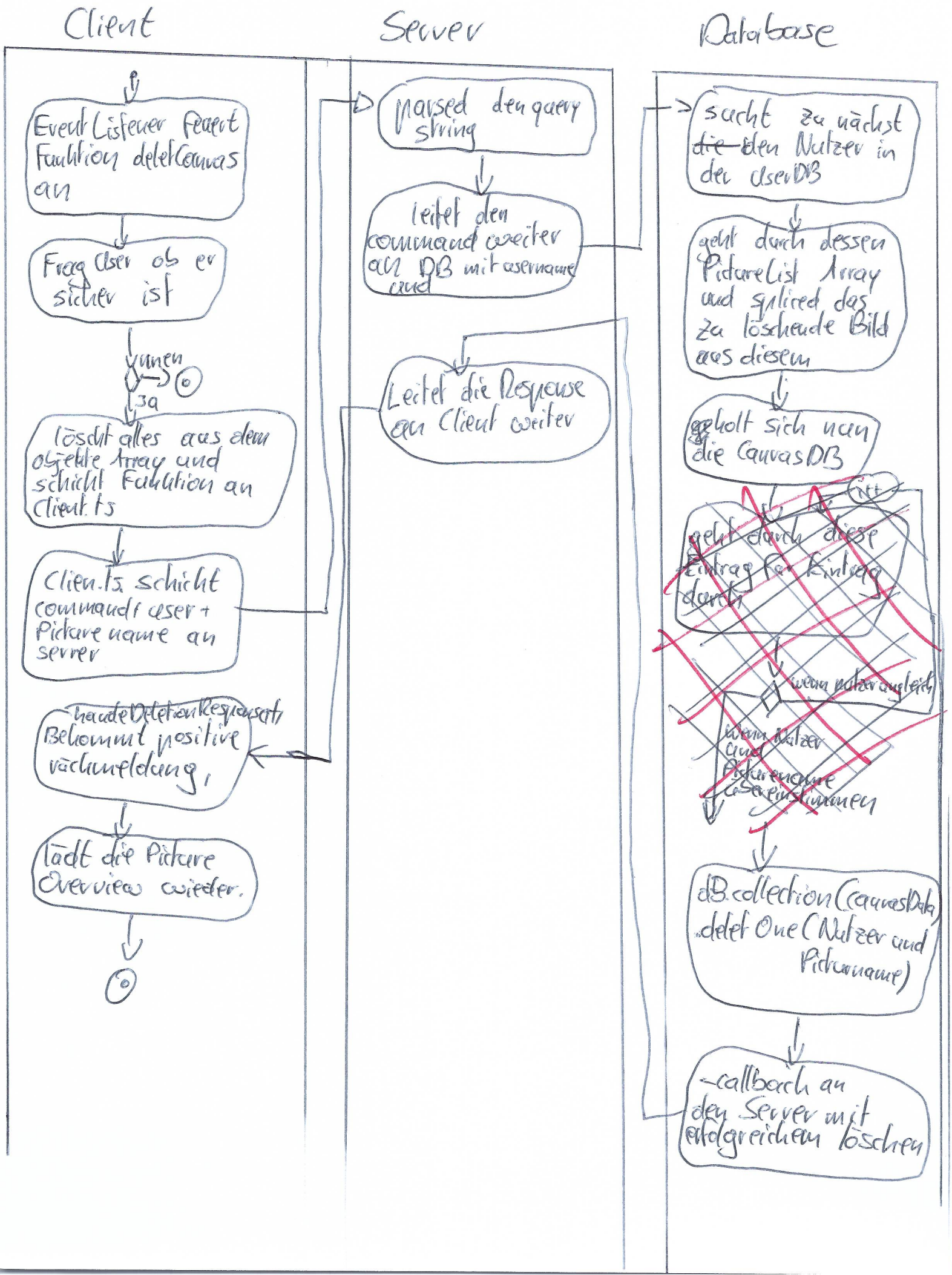
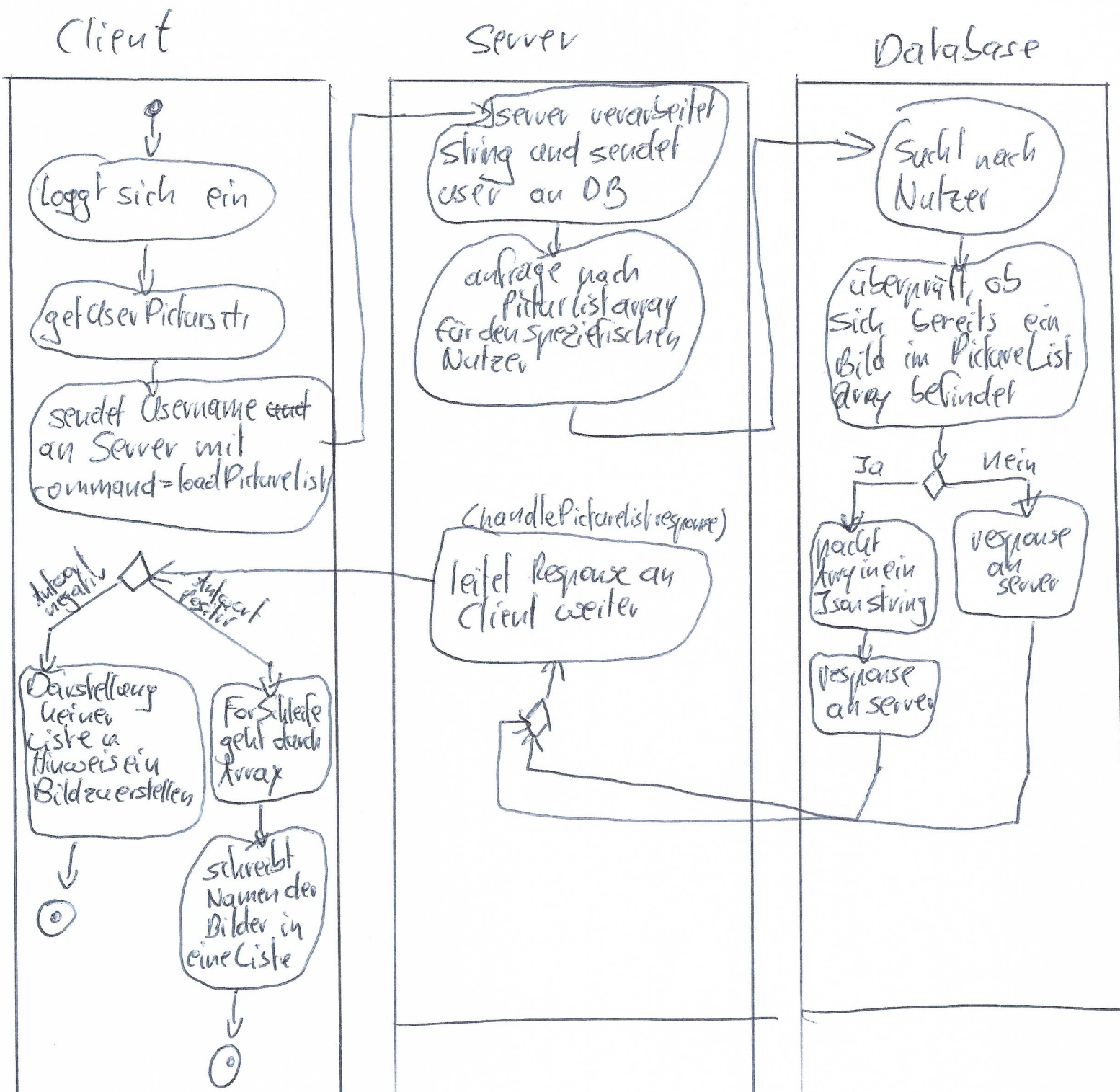
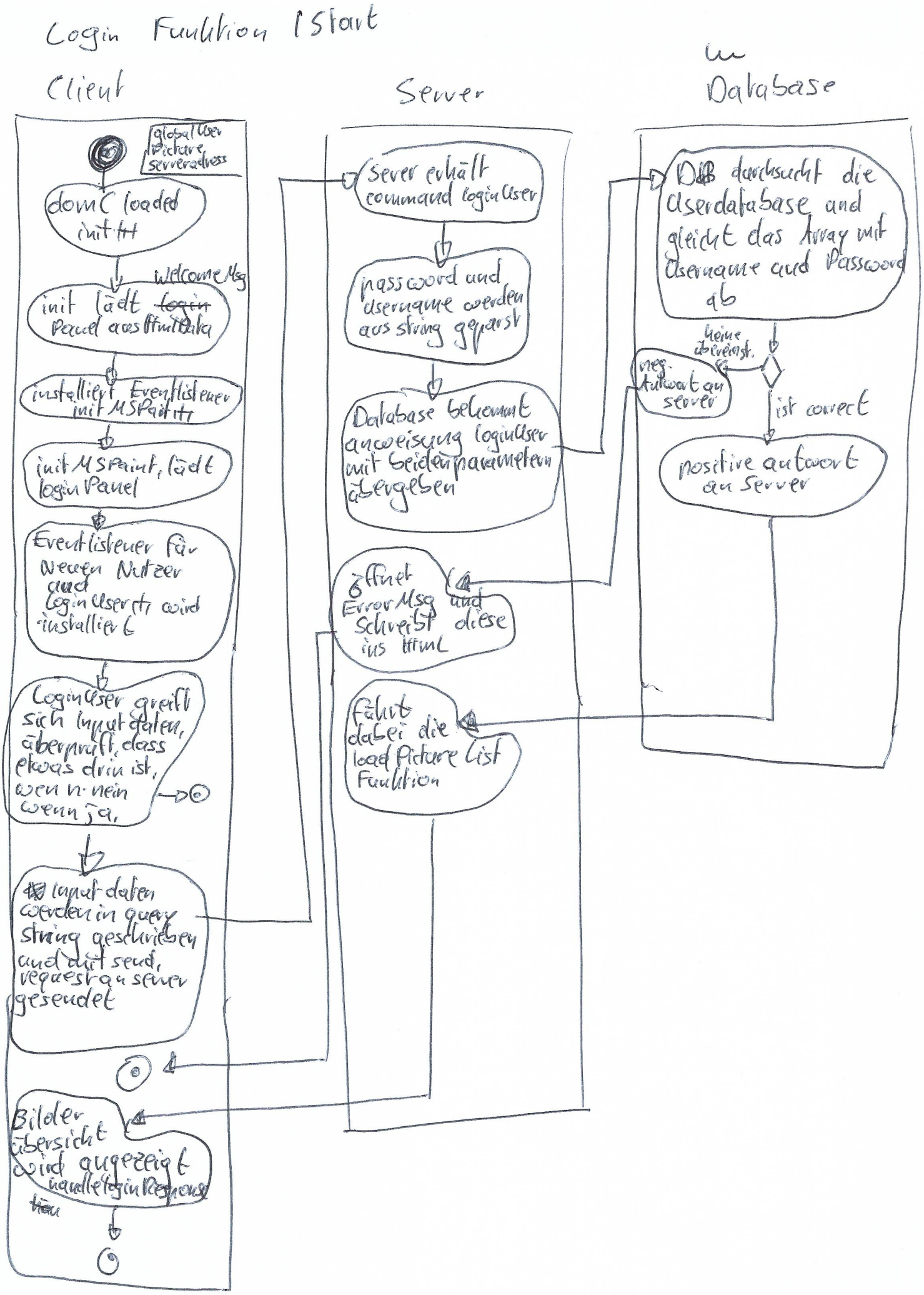


Bild übersichts Liste laden




Login Funktion Start
Client
Server
Database
global User Picture serveradress
domC loaded init ttt
welcome Msg
init lädt login Panel aus ttml Data
installiert EventListener init MSPaint ttt
init MSpaint, lädt login Panel
EventListener für Neuen Nutzer und Login User ttt wird installiert
LoginUser greift sich Input daten, überprüft, dass etwas drin ist, wenn nein wenn ja,
Input daten werden in query string geschrieben und mit send, request an server gesendet
Bilder übersicht wird angezeigt inaktiv login Response
Server erhält command loginUser
password und username werden aus string geparst
Database bekommt anweisung loginUser mit beiden parametern übergeben
öffnet ErrorMsg und schreibt diese ins ttml
fährt dabei die load Picture List Funktion
DB durchsucht die Userdatabase und gleicht das Array mit Username und Password ab
keine übereinst.
neg. Antwort an server
ist correct
positive antwort an Server

Neuen Nutzer registrieren

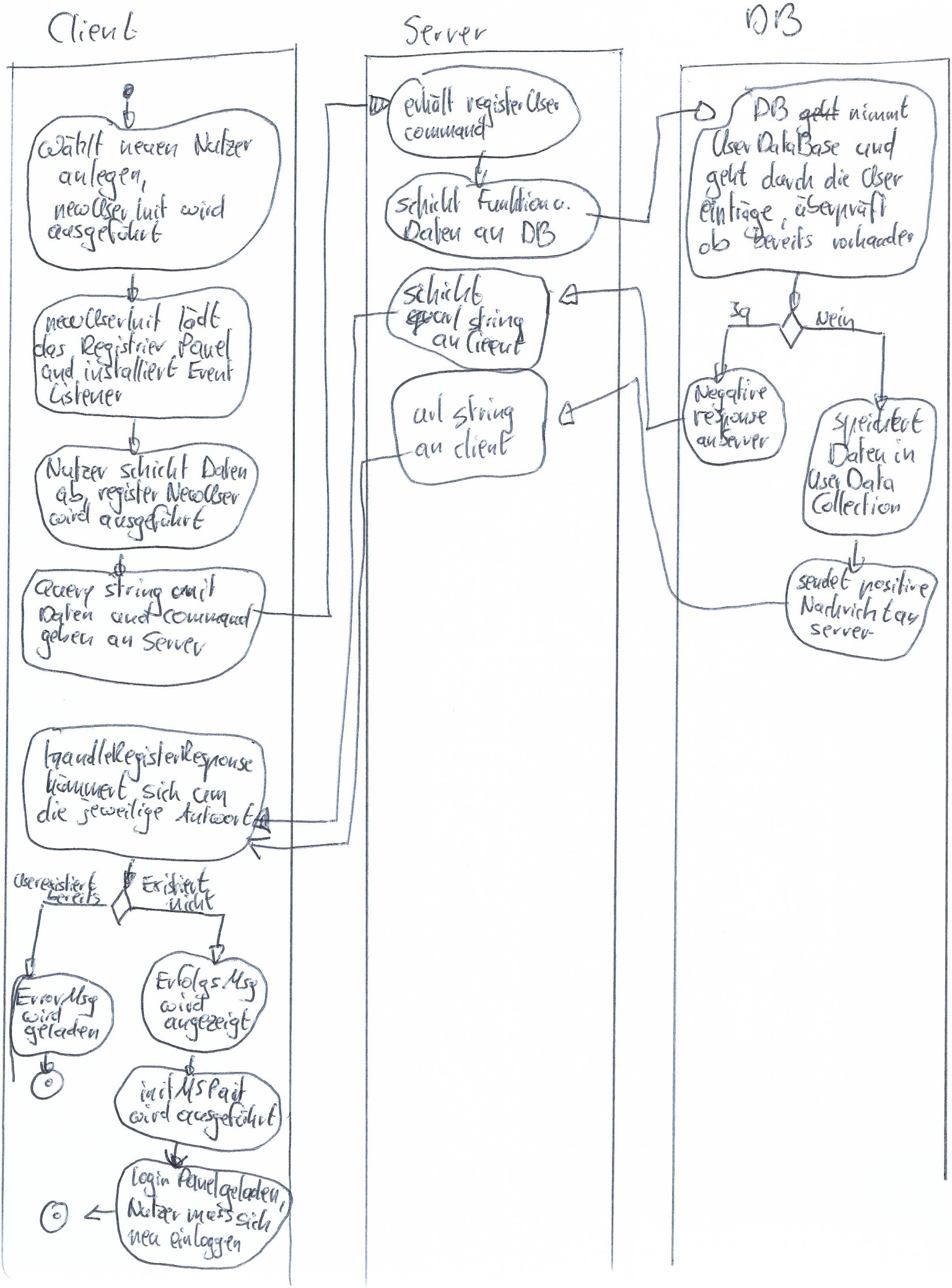


Bild anlegen / Canvas erstellen

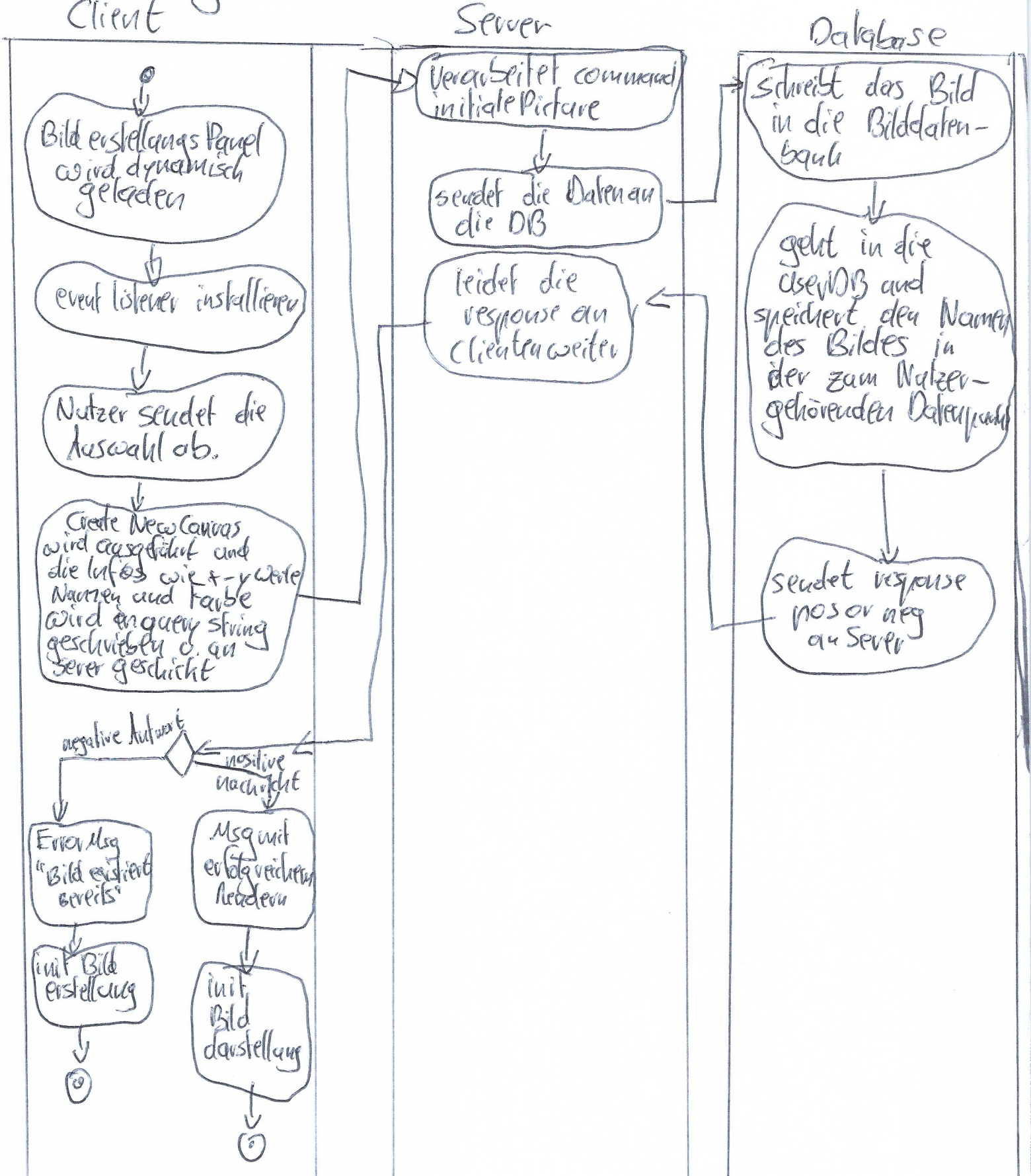


Bild für das erstemal erstellen und laden:

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

globale Variablen

> click: createNewCanvas

```
let ve: string = "#"
let query :string = "command = initialePicture"
let inputs: HTMLCollection
```

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

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

restliche inputs mit den jeweiligen queryKeys in der query schreiben und die globalen Variablen

```
query += "canvasColor=" + queryColor
```

```
send Request (query, handleNewCanvasResponse)
```

handleNewCanvasResponse

Wenn der Server "save Positive" weitergeleitet hat, wird initRenderCanvas ausgeführt

initRenderNewCanvas

innerHtml wird ersetzt durch html[Data][“mainCanvasPanel”]

```
initCanvas()
```

in der canvas.ts

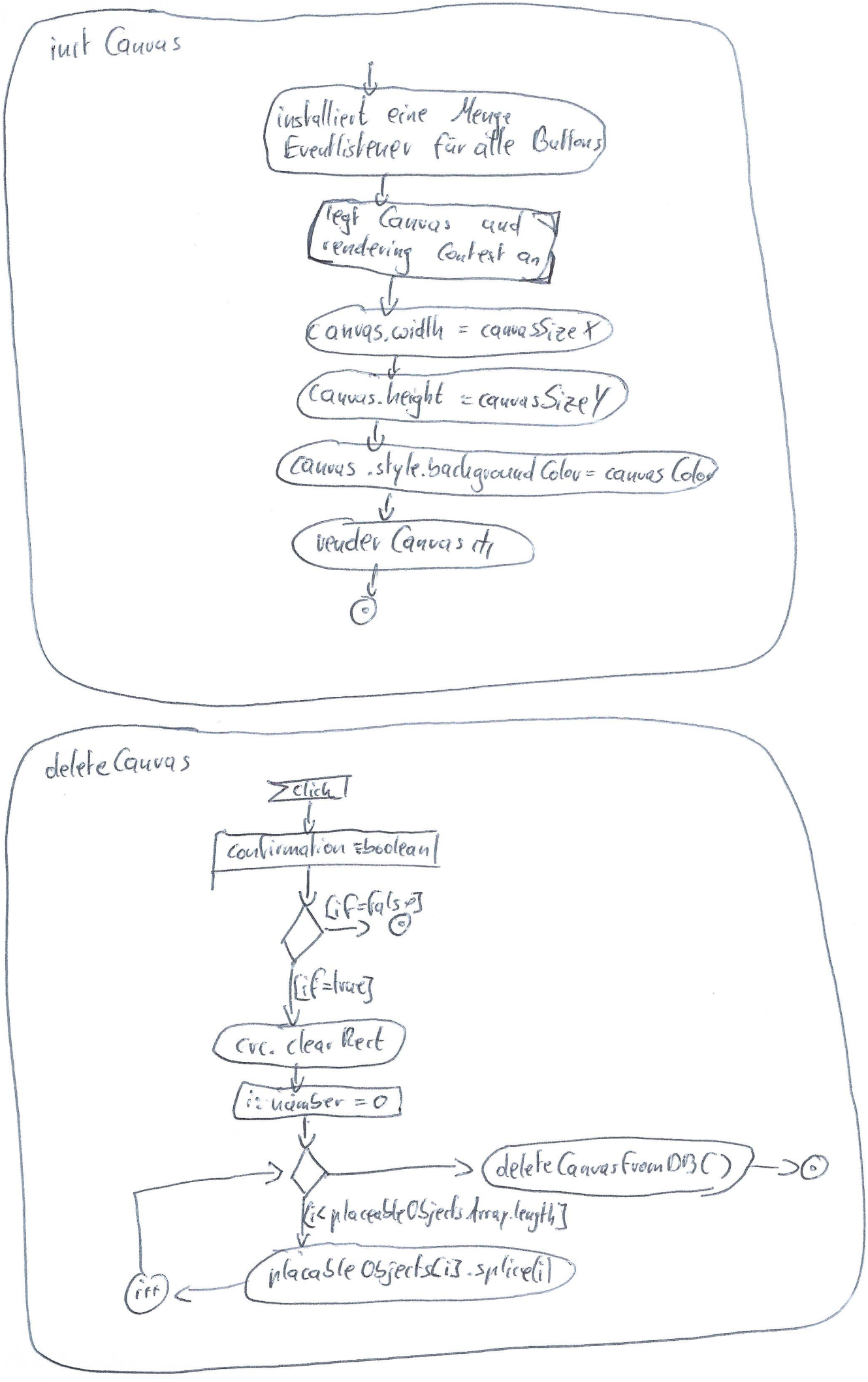
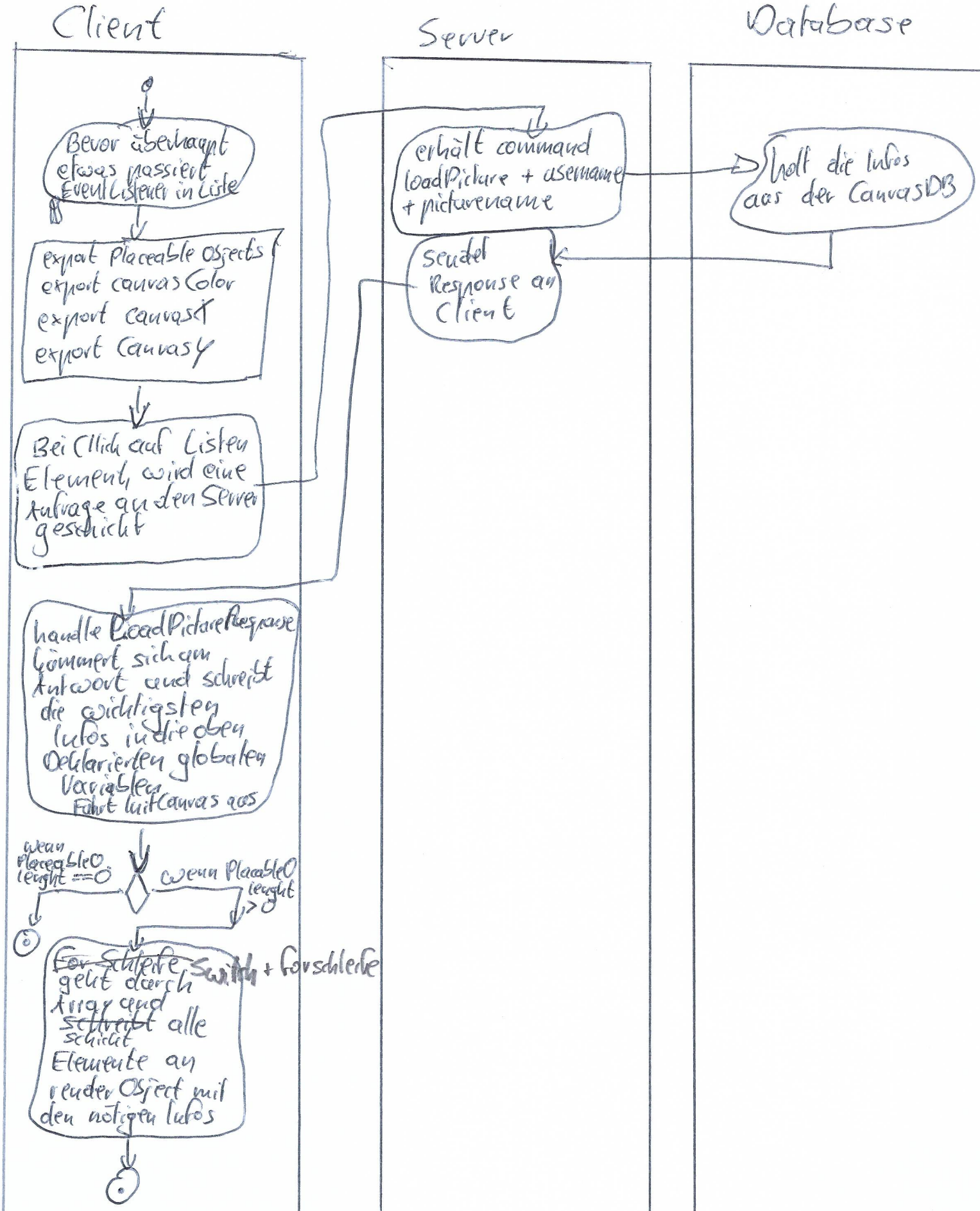

init Canvas
installiert eine Menge Eventlistener für alle Buttons
legt Canvas und rendering Context an
canvas.width = canvasSizeX
canvas.height = canvasSizeY
canvas.style.backgroundColor = canvasColor
render Canvas ()
deleteCanvas
click
confirmation = boolean
[if=false]
[if=true]
ctx.clearRect
i: number = 0
deleteCanvasFromDB()
[i<placeableObjects.length]
placableObjects[i].splice(i)
i++

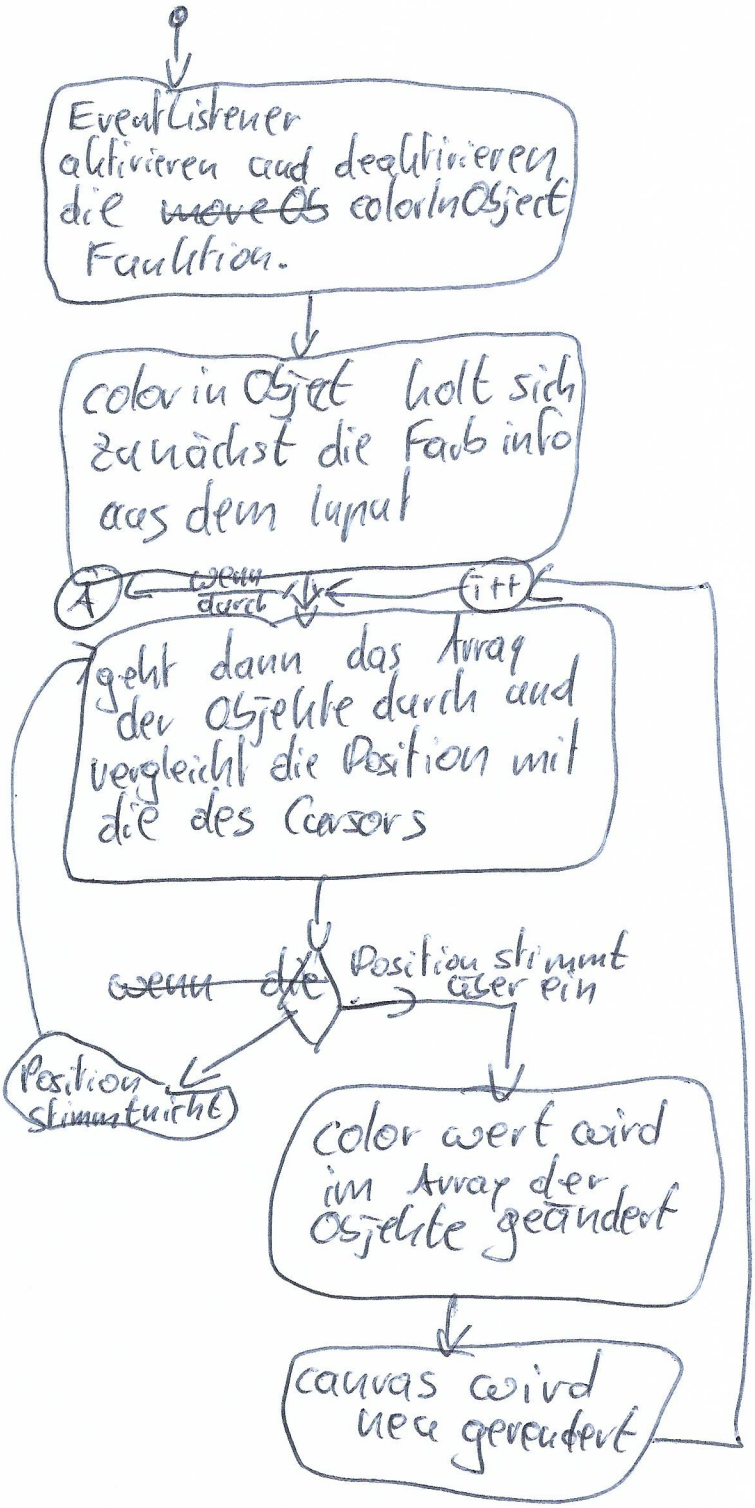
Bild erneut bearbeiten:

⇒ woher weiß der Button, welche ID gedrückt wurde?



Objekte Farben:

⇒ im großen und ganzen die Bewegen Funktion

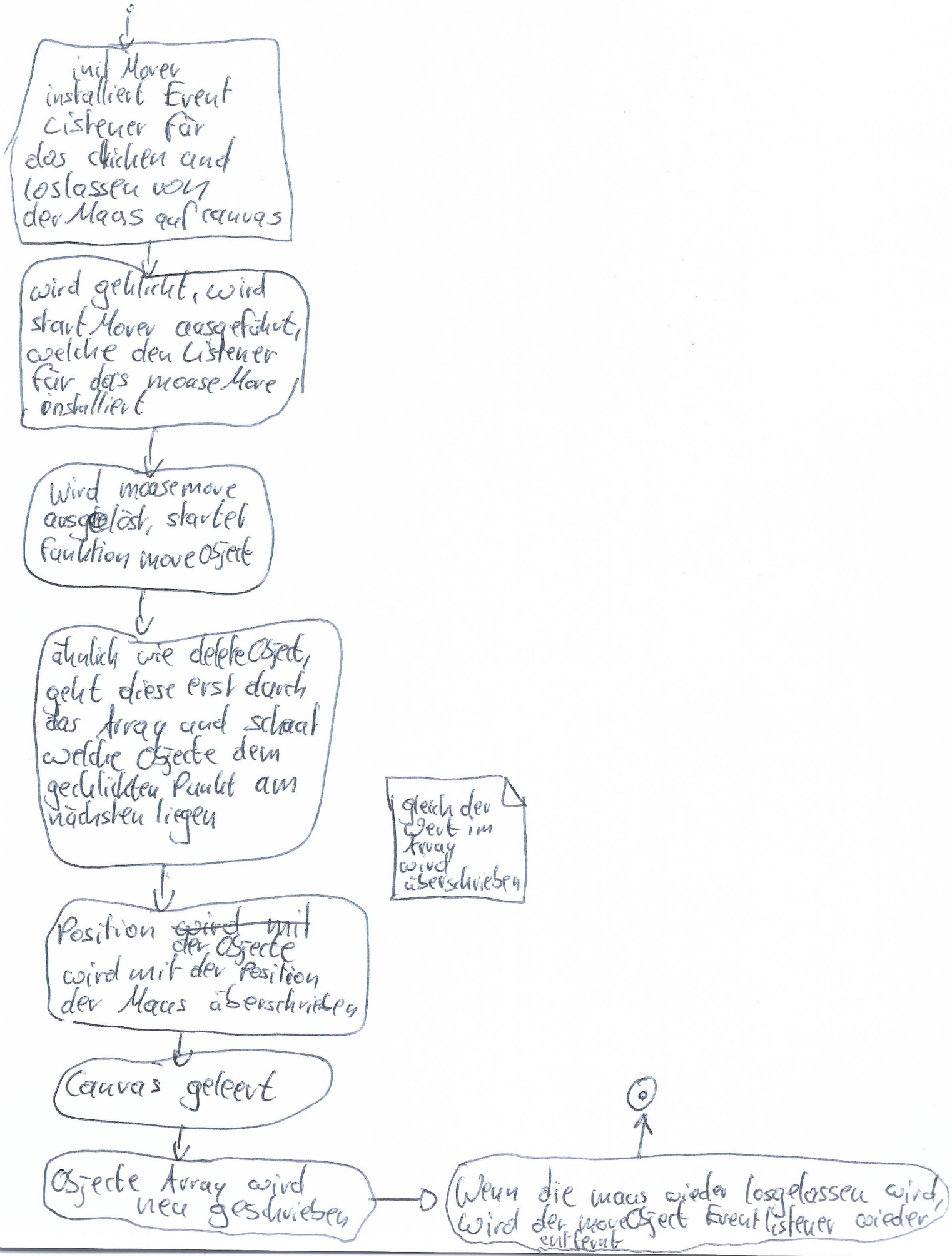


Änderung des ~~Arrays~~ Klassenes

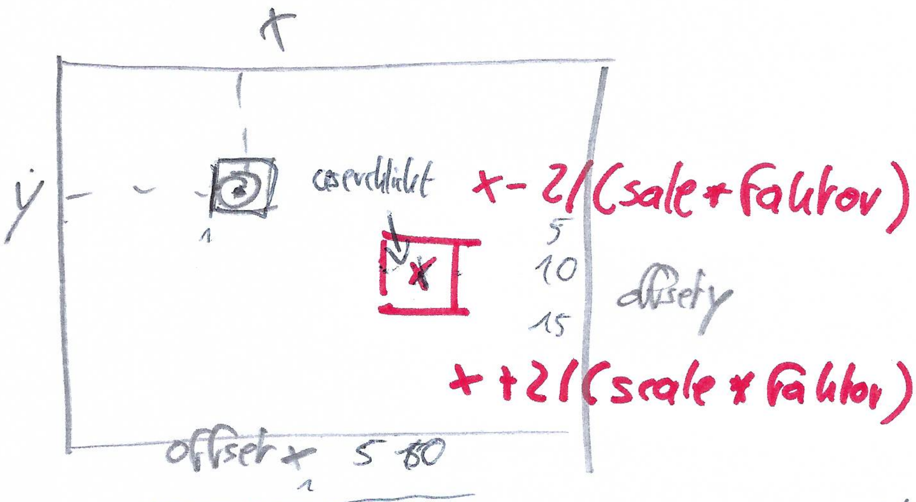
⇒ selber Ansatz, nur wird die type Variable im Objekte Array am geschrieben.

Bewegen von Objekten

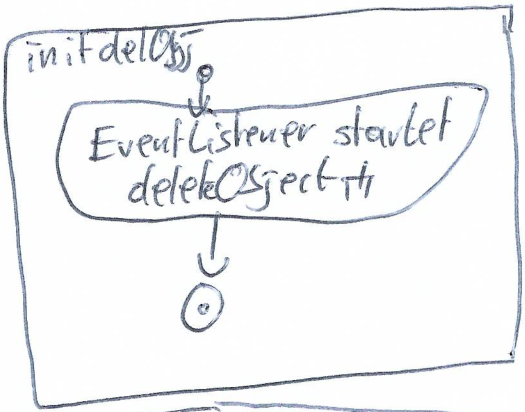
→ Grundarbeit kommt vom löschen



Objekt von Canvas löschen.

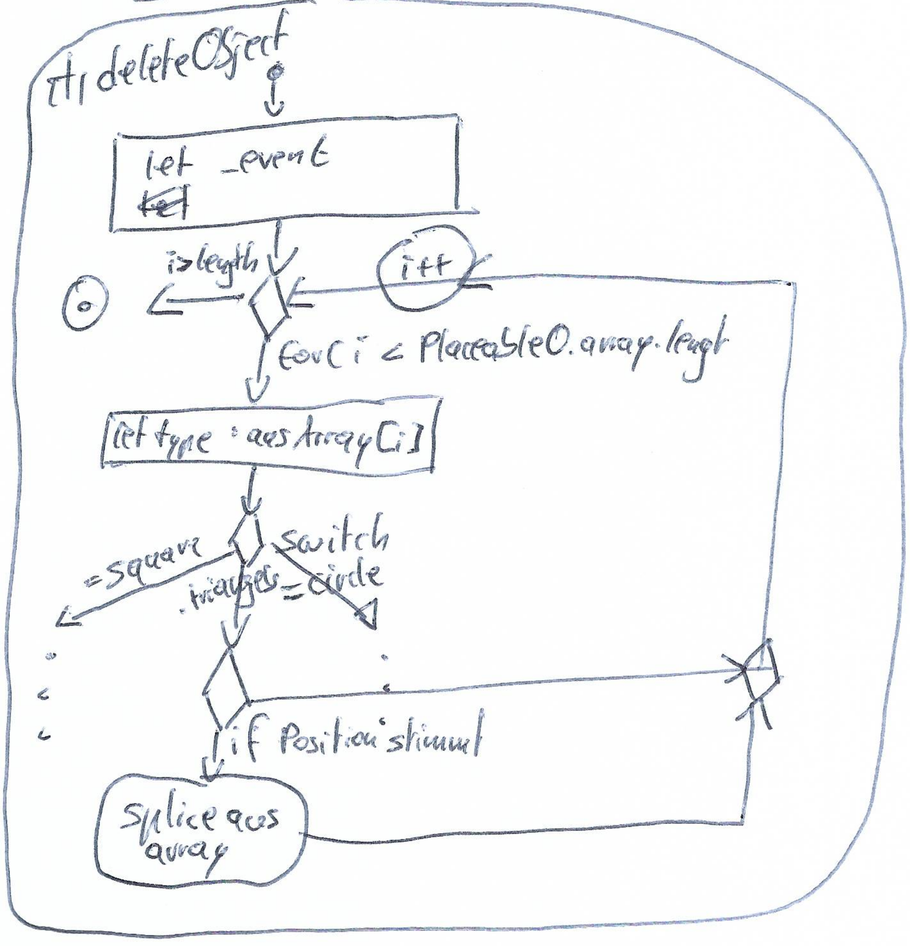


Für x: array.x − scale×factor/2 >= userx
array.x + '' <= userx
⋮



→ in if clauses berechnet es komischerweise falsch
⇒ Lösung zunächst in Variablen berechnen

ifsizeX
ifsizeXm
ifsizeY
ifsizeYm



z.B.

x 54 88

userx: 55

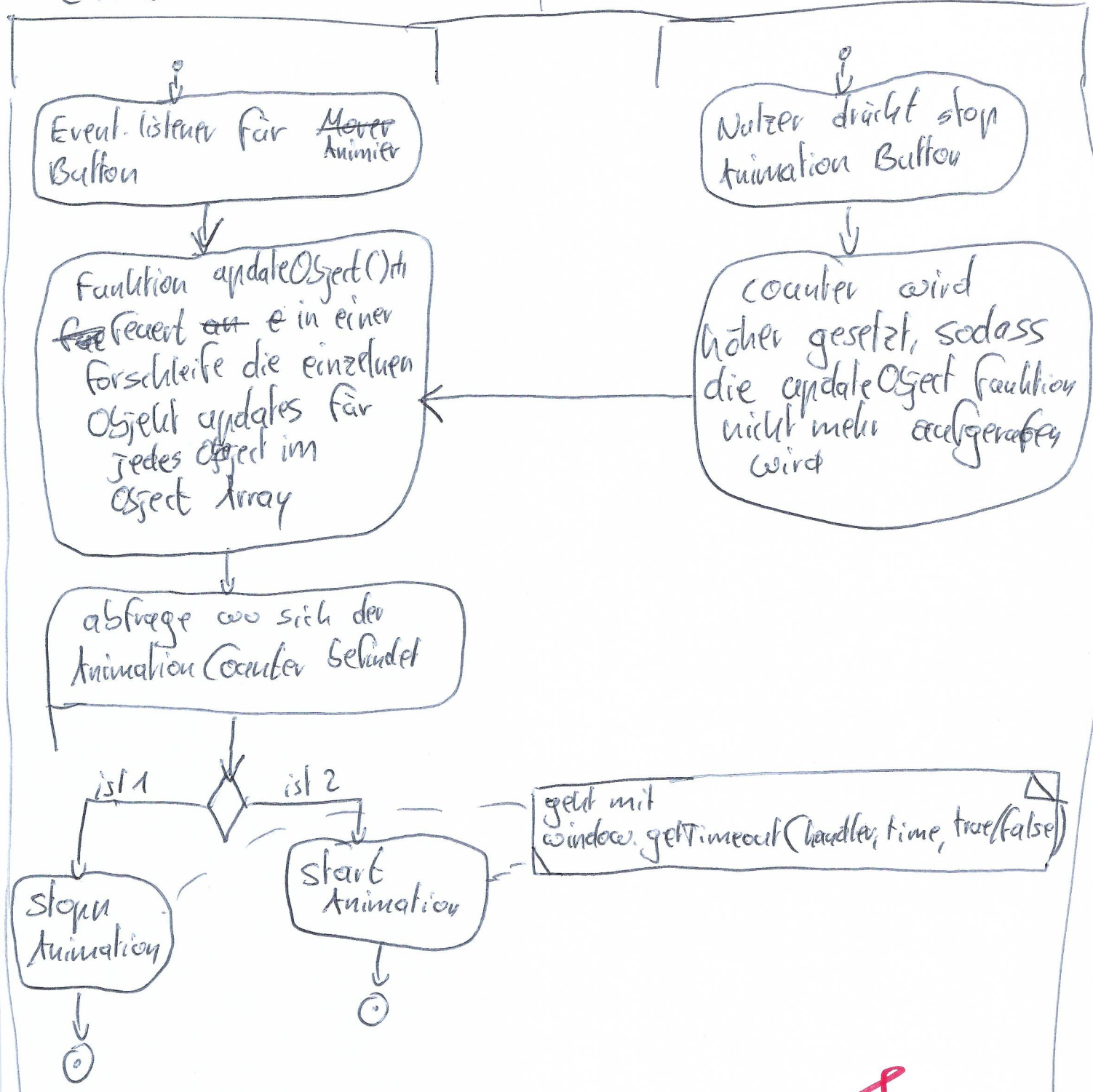
x 62
y 53

Bild animieren lassen

canvas.ts

Bild Animation pausieren

canvas.ts



- das Bild muss nach Bewegung neu gerendert werden dafür muss das Object 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

-_-_-_-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?

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 installes compiler, which complies the
  TypeScript your about to change

Step by Step

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

2-in the folder 2_Progamm > 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 in side the Cluster

6-create two collections with the names:
  Userdatabase
  canvasDatabase

7-push all changed to gitHub

8-deploy your heroku app

You should be ready to go