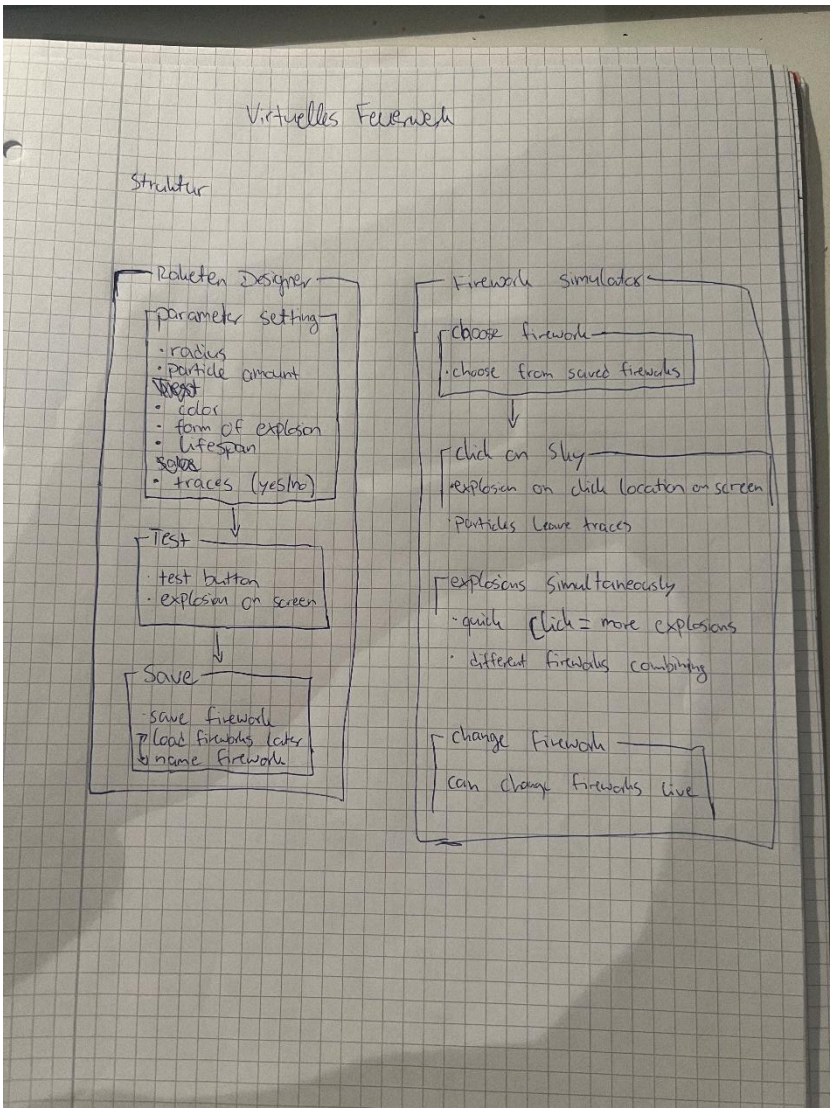
Endabgabe Feuerwerks-Simulation EIA2 WISE24/25

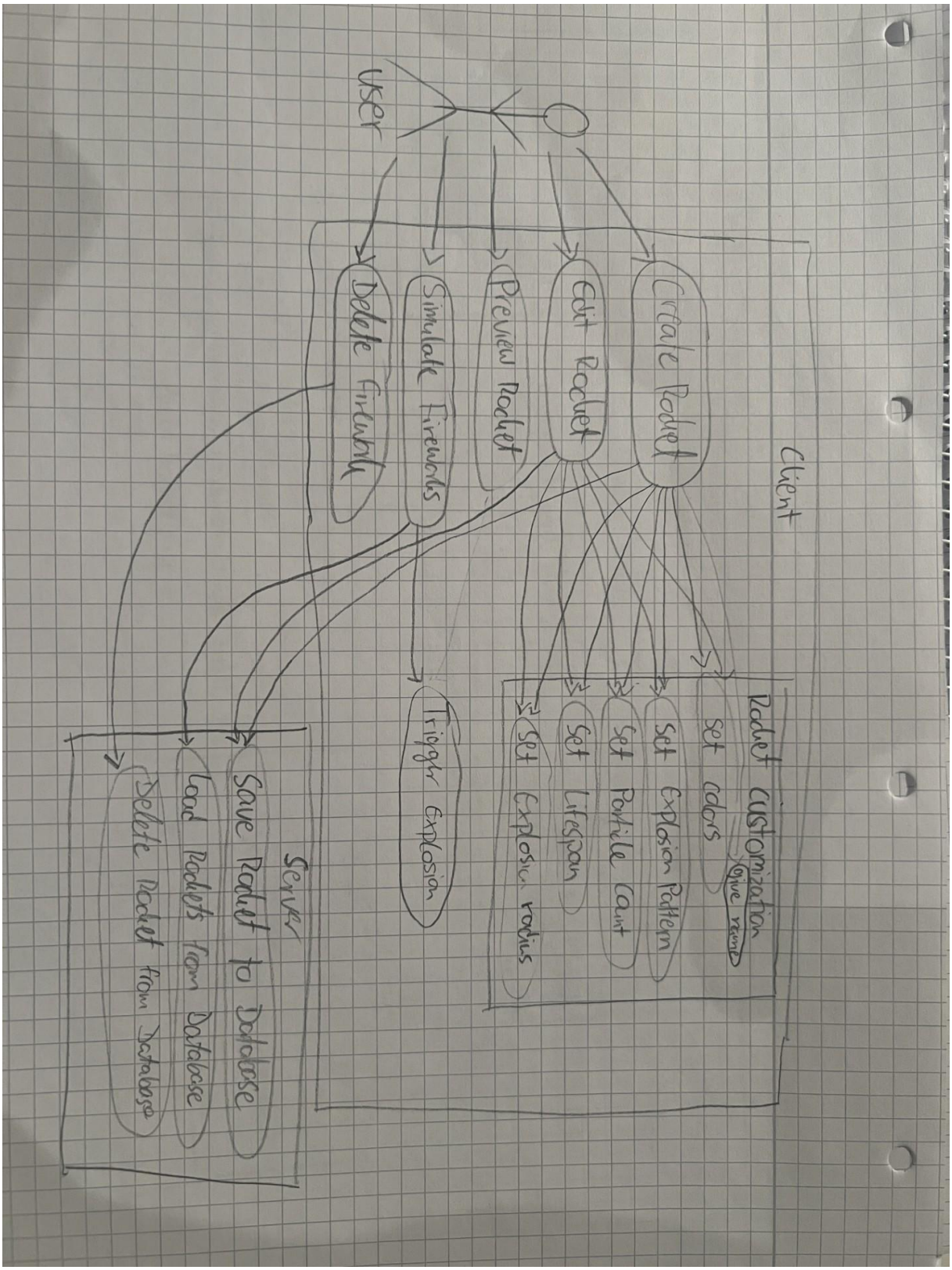
Maximilian Duczmal MKB2 Matrikelnr: 277173

Die digitale Feuerwerksanwendung ermöglicht es Nutzern, eigene Feuerwerksraketen zu gestalten, zu speichern und auf einem virtuellen Nachthimmel abzubrennen. Die Anwendung kombiniert eine benutzerfreundliche Oberfläche mit einer dynamischen Canvas-Animation, um ein interaktives Feuerwerkserlebnis zu bieten.

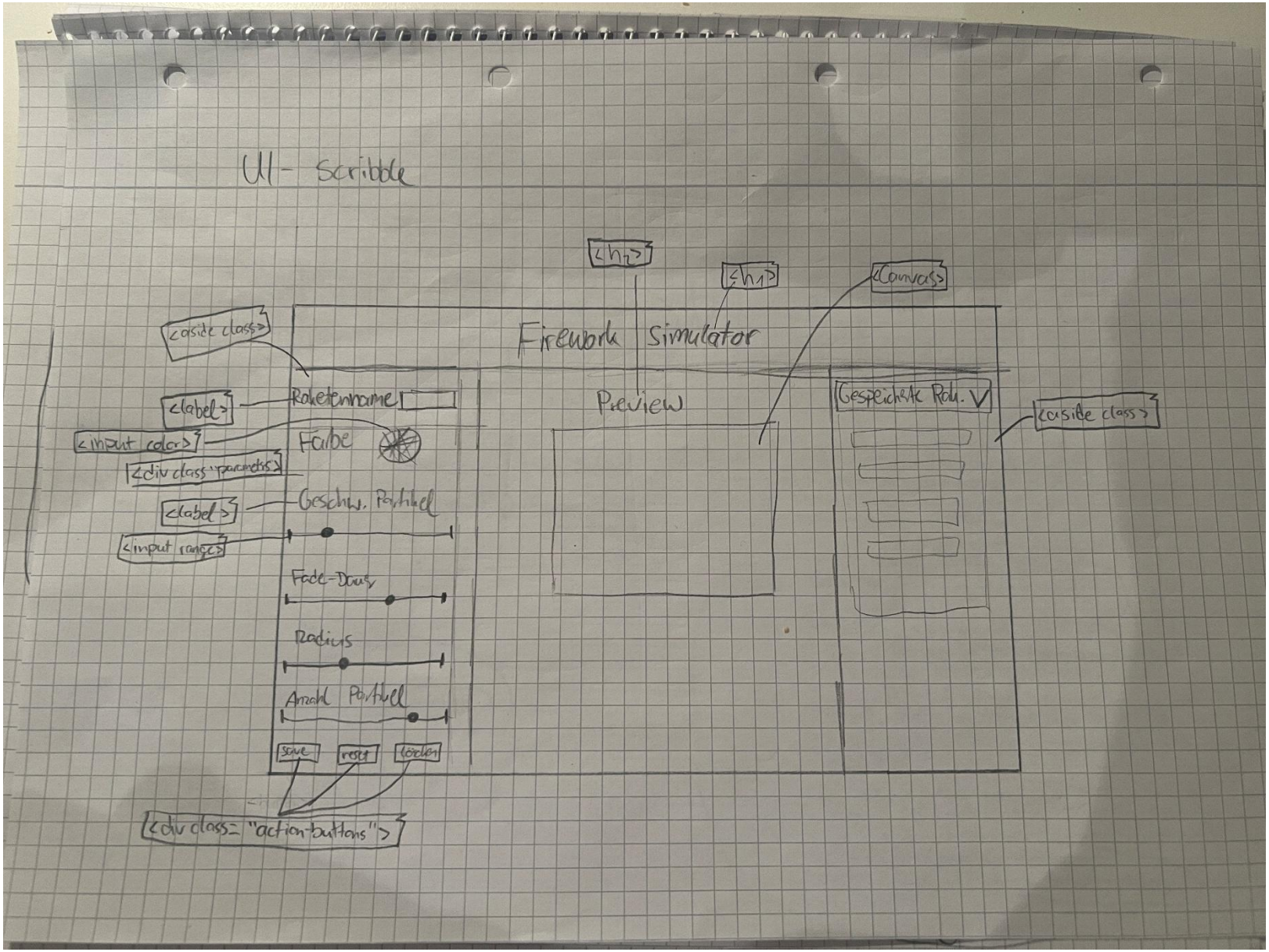
Strukturplan:



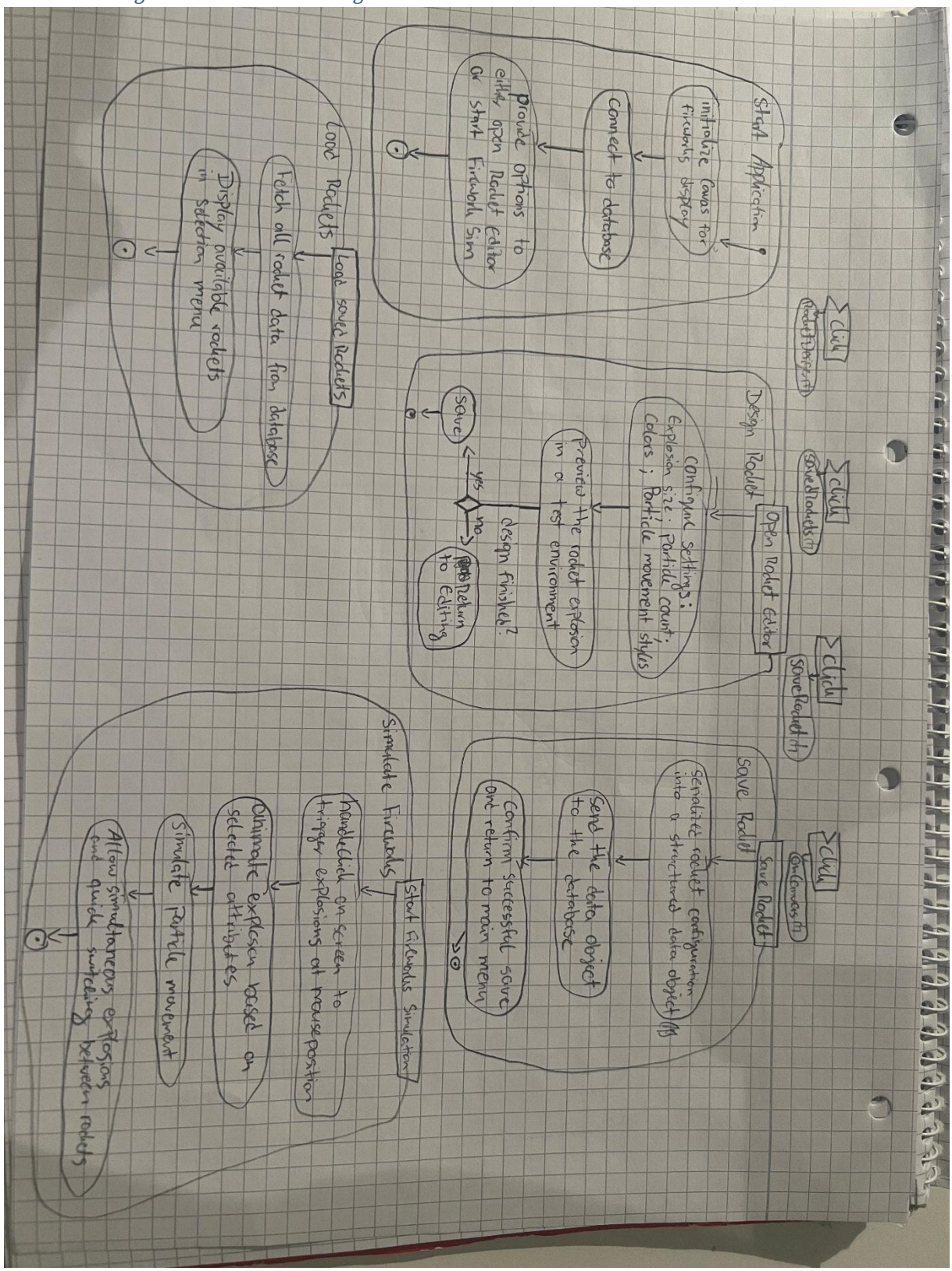
Use Case Diagramm:

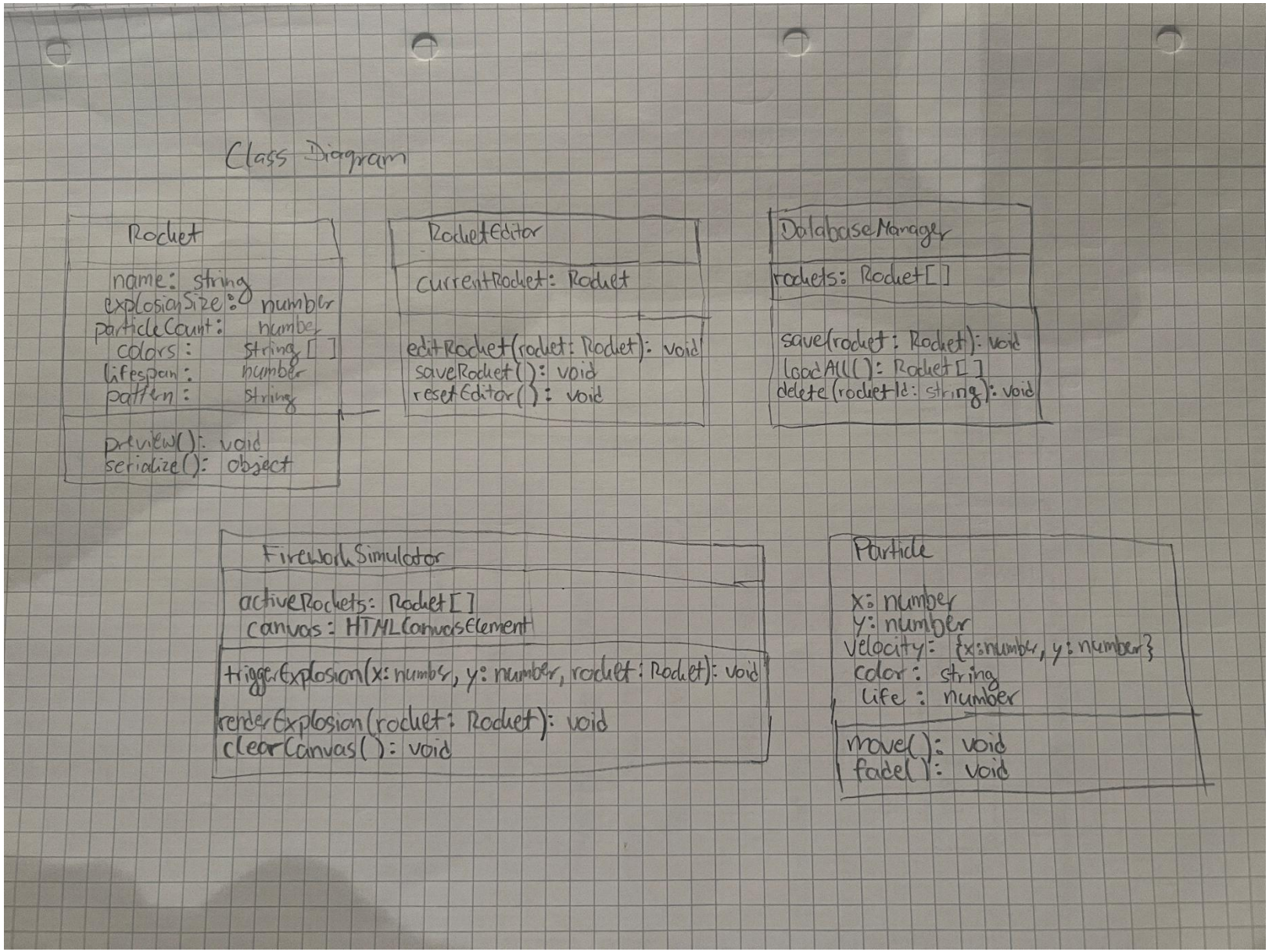


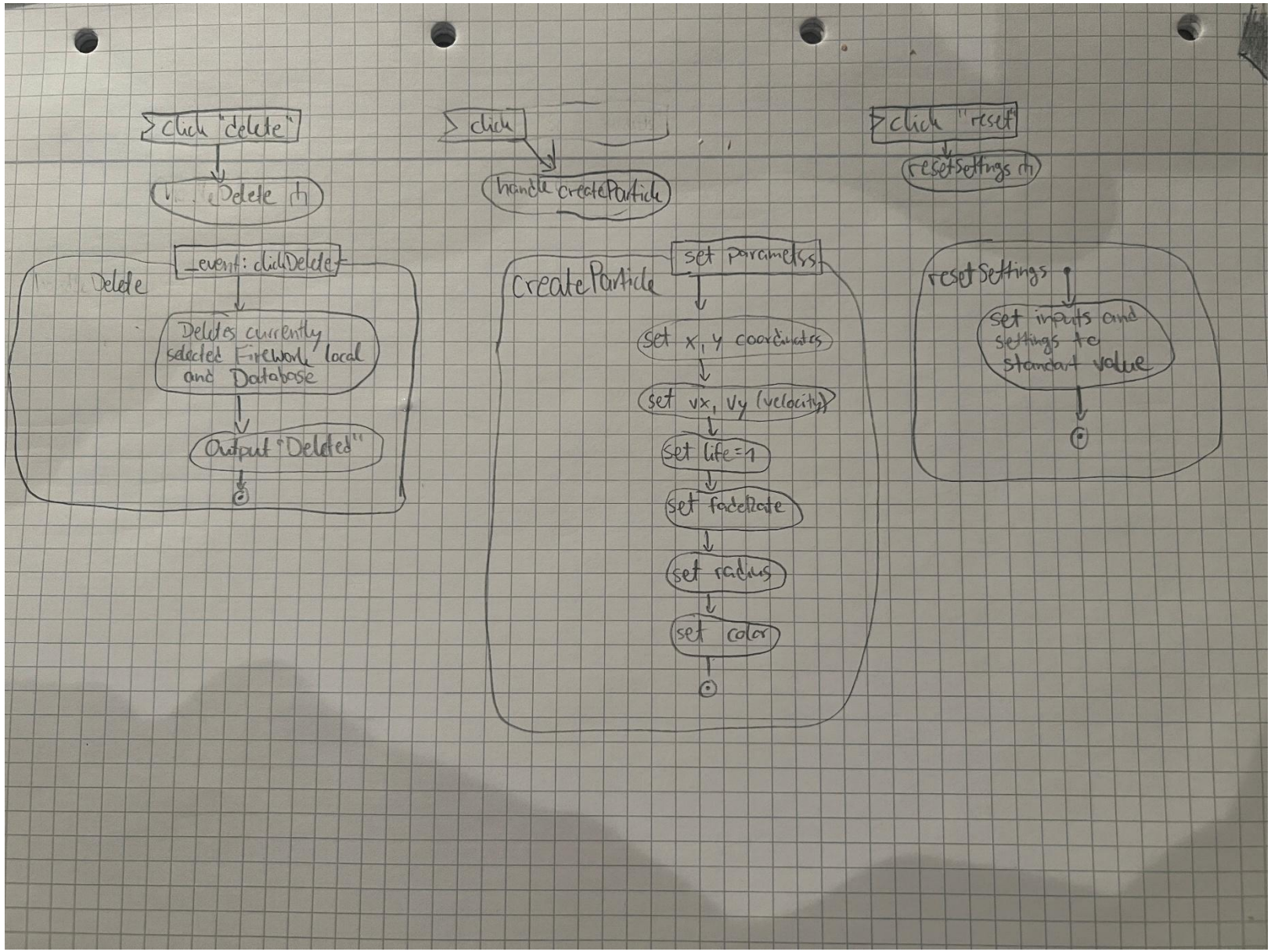
UI Scribble:


UI - Scribble
<h2>
<h1>
<canvas>
<aside class>
Firework Simulator
Preview
Gespeicherte Rak.
<aside class>
<label>
Raketenname
<input color>
Farbe
<div class "parameters">
<label>
Geschw. Partikel
<input range>
Fade-Dauer
Radius
Anzahl Partikel
save
reset
löschen
>

Aktivitäts-Diagramm und Klassendiagramm:




Class Diagram
Rocket
name: string
explosionSize: number
particleCount: number
colors: string[]
lifespan: number
pattern: string
preview(): void
serialize(): object
RocketEditor
currentRocket: Rocket
editRocket(rocket: Rocket): void
saveRocket(): void
resetEditor(): void
DatabaseManager
rockets: Rocket[]
save(rocket: Rocket): void
loadAll(): Rocket[]
delete(rocketId: string): void
FireworkSimulator
activeRockets: Rocket[]
canvas: HTMLCanvasElement
triggerExplosion(x: number, y: number, rocket: Rocket): void
renderExplosion(rocket: Rocket): void
clearCanvas(): void
Particle
x: number
y: number
velocity: {x: number, y: number}
color: string
life: number
move(): void
fade(): void


click "delete"
Delete
_event: clickDelete
Delete
Deletes currently selected Firework local and Database
Output "Deleted"
click
handle createParticle
set parameters
createParticle
set x, y coordinates
set vx, vy (velocity)
Set life=1
set fadeRate
set radius
set color
click "reset"
resetSettings
resetSettings
Set inputs and settings to standart value