Produktrapport Kortklubben



Indhold

[Produktrapport Kortklubben 1](#_Toc105504027)

[Titelblad 2](#_Toc105504028)

[Kravspecifikation 4](#_Toc105504029)

[Brugervejledning 4](#_Toc105504030)

[Server 4](#_Toc105504031)

[Root server: 4](#_Toc105504032)

[Database (Virtuelle server 1): 4](#_Toc105504033)

[Game (Virtuelle server 2): 4](#_Toc105504034)

[Overvejelser/beslutninger 5](#_Toc105504035)

# Kravspecifikation

Der er taget udgangspunkt efter FURPS og er den der vil blive refereret til igennem dokumentet.

Der er blevet lavet lidt om på use-case 4 ift. at se sin historik. I stedet så er der et leaderboard, hvor man ville kunne se sine egne stats 🡪 vundet, tabt og spil spillet, samt top x spillere.

# Brugervejledning

## Server

Der anvendes en server, som har 2 virtuelle servere til at administrere og køre vores system. Her er følgende informationer:

### Root server:

Type: x64-baseret computer

* Fabrikant: LENOVO
* Model: 10M9000CMT
* CPU: 4 x Intel® Core™ i7-7700 CPU @ 3.60GHz
* Hukommelse: 32 GB
* Opbevaring: 350 GB

### Database (Virtuelle server 1):

* Host navn: ZBC-RG-DBSERVER-2022
* Version: Microsoft Windows Server 2022 Standard
* Type: x64-baseret computer
* CPU: 2 x Intel® Core™ i7-7700 CPU @ 3.60GHz
* Hukommelse: 8 GB
* Opbevaring: 90 GB

### Game (Virtuelle server 2):

* Host navn: ZBC-RG-GAMESERVER-2022
* Version: Microsoft Windows Server 2022 Standard
* Type: x64-baseret computer
* CPU: 2 x Intel® Core™ i7-7700 CPU @ 3.60GHz
* Hukommelse: 8 GB
* Opbevaring: 90 GB

På **Database serveren** er der lagt en SSMS (SQL Server Management Studio) på, samt SCCM (System Center Configuration Manager). Derudover er der åbnet for to port. En på **Inbound Rules** og en på **Outbound Rules**.

#### Inbound rule:

* Type: TCP
* Lokale port: 1433
* Remote port: Alle porte

#### Outbound Rule:

* Type: UDP
* Lokale port: 1434
* Remote port: 1434

De 2 regler bruges til at tillade kommunikation til/fra database i netværket.

På **Game serveren** er der lagt Node.js og PM2 (Process Management).

# Overvejelser/beslutninger

I starten var der taget beslutning om at bruge Tomcat til websocket, men er så herefter blevet skiftet over til Node, årsagen for dette var at Tomcat forvoldte en del modgang med Java.

Derudover så har vi valgt at gå fra at lave vores engine i Java, til at lave den i TypeScript/JavaScript, af den årsag, at det kræver Java Enterprise som koster penge, for at kunne håndtere API kald og diverse web-related funktioner.