1 Iteration

Systemet lader spiller vælge om der ønskes at starte en ny hero eller loades eksisterene. Hvis der ønskes ny spørges der efter navn på ny hero og der oprettes en ny hero med start stats 2 styrke 10 HP level 1 og 0 xp. Hvis der ønskes load læses en .txt vil med gemte heros i. Alle heros gemt i filen bliver vist, og spiller kan vælge hvilken at spille som. Her efter vises nogle standart fjender som spiller kan vælge at kæmpe imod. Hvis fjenden vinder kan spiller vælge en ny fjende at kæmpe imod. Hvis spiller vinder tilføjes xp reward til hero. Og der ses om hero har xp nok til at lvl op hvis ja, så lvl hero op og modtager 2 ekstra hp og 1 ekstra styrke. Spiller har også mulighed for at lukke spillet ved at trykke q. Når spillet lukkes gemmes heros data til .txt filen, der undersøgs først om der eksistere en hero med nuværne hero navn, hvis ja opdateres dataen i .txt filen. Ellers tilføjes der ny data til .txt filen.

1.1 Use cases

- **Start nyt spil** - Spiller vælger ny hero og indtaster ønsket navn, der oprettes en ny hero med det givne navn og som har en styrke på 2 og 10 HP, i lvl 1 med 0 xp.
- **Load Hero** - Spiller vælger load hero, spiller vælger hvilken af de gemte heros der ønskes at blive spillet med. Og denne heros data loades ind i spillet.
- **Vælg fjende** - Spiller vælger en fjende at kæmpe imod, fra listen over de mulige fjender. Og spiller føres videre til kampen.
- **Kæmp mod fjende** - Når spiller har valgt en fjende, kæmper spillet automatisk mod fjenden, slag for slag mellem hero og valgt fjende. Resultatet af kampen vises til spiller. Og eventuelt optjent xp gives til spiller.
- **Forlad spil** - Spiller skal have mulighed for at kunne lukke spillet fra hovedmenuen. Når spillet lukkes bliver data for hero i den aktuelle session gemt/opdateret, til næste gang spillet startes.

1.2 Use case diagram

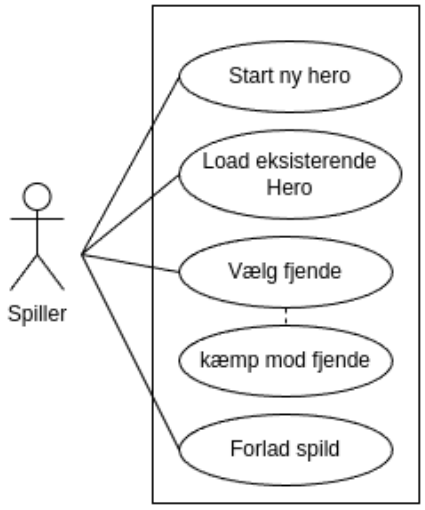

Figure 1: Use case diagram

1.3 Domain model

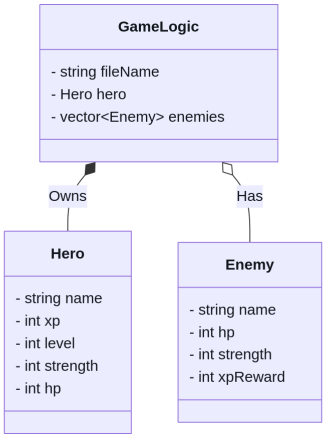

Figure 2: Domain model

1.4 Sekvens diagramer

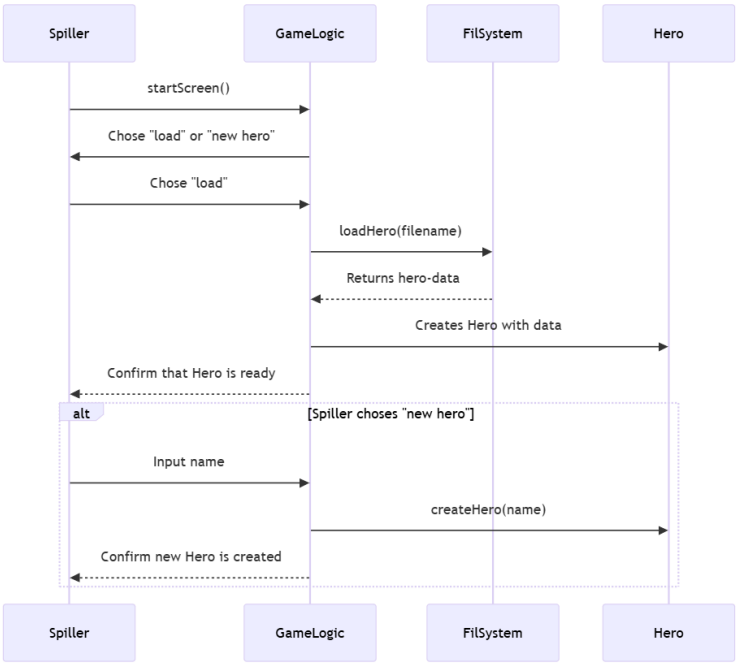

Figure 3: Sekvens diagaram

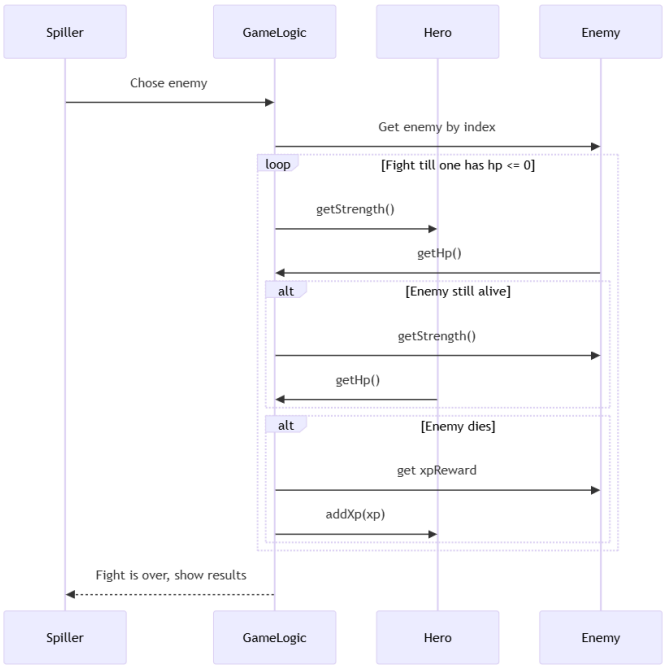

Figure 4: Sekvens diagram

1.5 UML class diagram

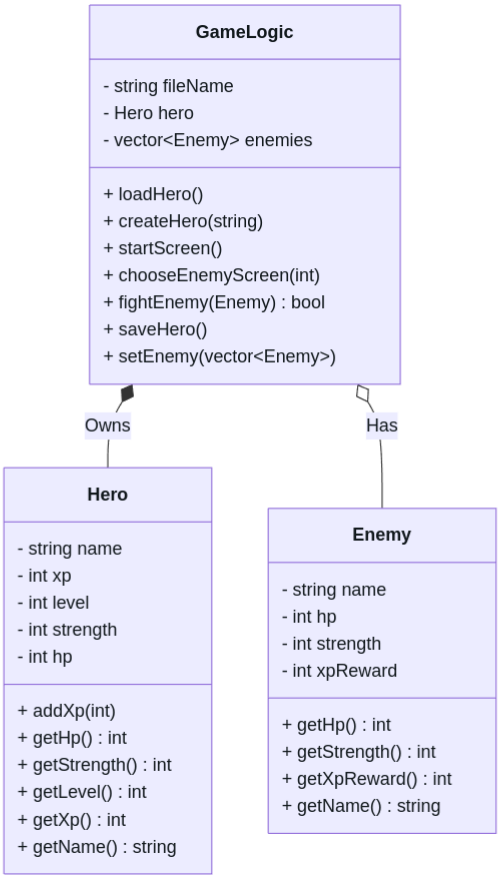

Figure 5: UML class diagram

2 Iteration

Systemet er nu istand til at generere grotter dynamisk med forskellige navne og fjender, som passer i styrke, til heros styrke og level. Hver grotte indeholder også guld efter sverheds grad som til deles hero når alle fjender i grotten er besejret. Det betyder at hver gang spillet startest bliver der genereret fjender som passer til heros styrke, så man kan blive ved med at spille spillet uden at hero bliver for overpowered til de fjender som er i spillet.

2.1 Use cases

- **Vælg grotte** - Spiller vælger en grotte at gå ind. I grotten er der en række automatisk genereret fjender. Spiller vælger fjende at kæmpe imod. Grotten skal også indholde en specifik mængde gold.
- **Bekæmp alle fjender i grotte** - Når spiller har vagt en grotte, skal spiller kæmpe mod alle fjender i den valgte grotte, en adgangen for at gennemføre grotten. Når grotten er gennemført tildeles spiller gold
- **Få tildelt gold** - Når spiller har gennemført en grotte, skal spilleren modtage gold fra grotten som belønning. Gold skal gemmes som en del af Heros data. Som kan gemmes og loades ved start sammen med hero.
- **Automatisk generer grotte** - Systemet skal automatisk generere grotter dynamisk som passer til valgte heros level. Resultatet er at der vises en række grotter med forskellige navne og fjender i. Med gold efter sværheds grad.
- **Automatisk generer fjender** - Systemet skal automatisk generere fjender til grotterne. Fjenderne skal laves dynamisk, så deres hp og styrke passer til heros level og sværhedsgraden af grotten. Fjendernes navne skal passe til grotte typen.

2.2 Use case diagram

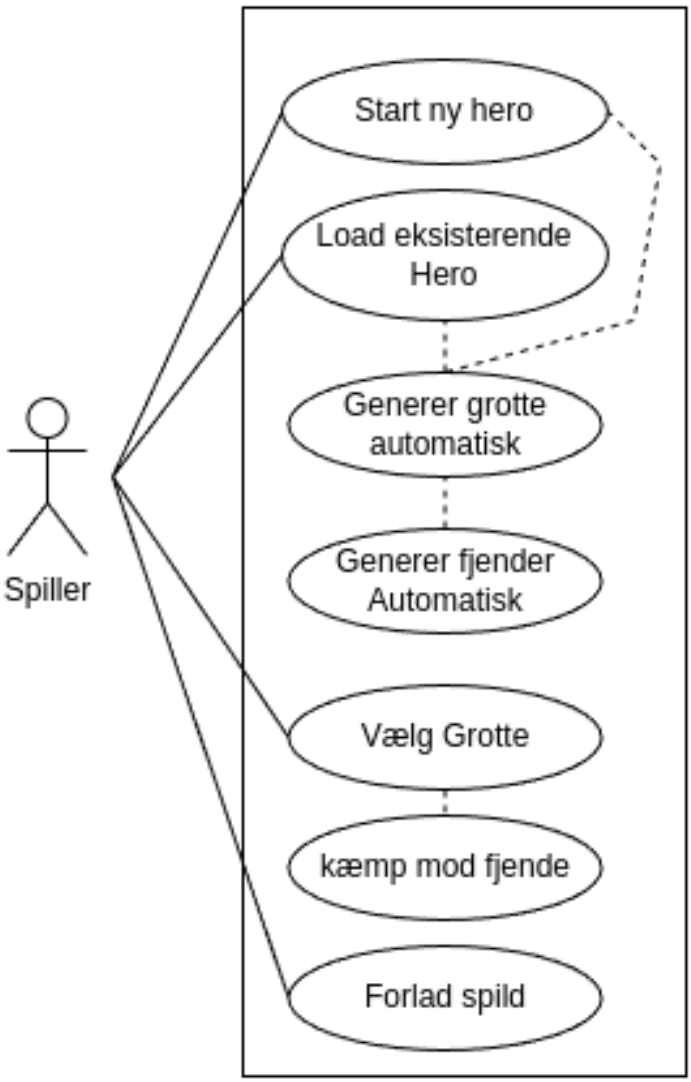

Figure 6: Use case diagram

2.3 Domain model

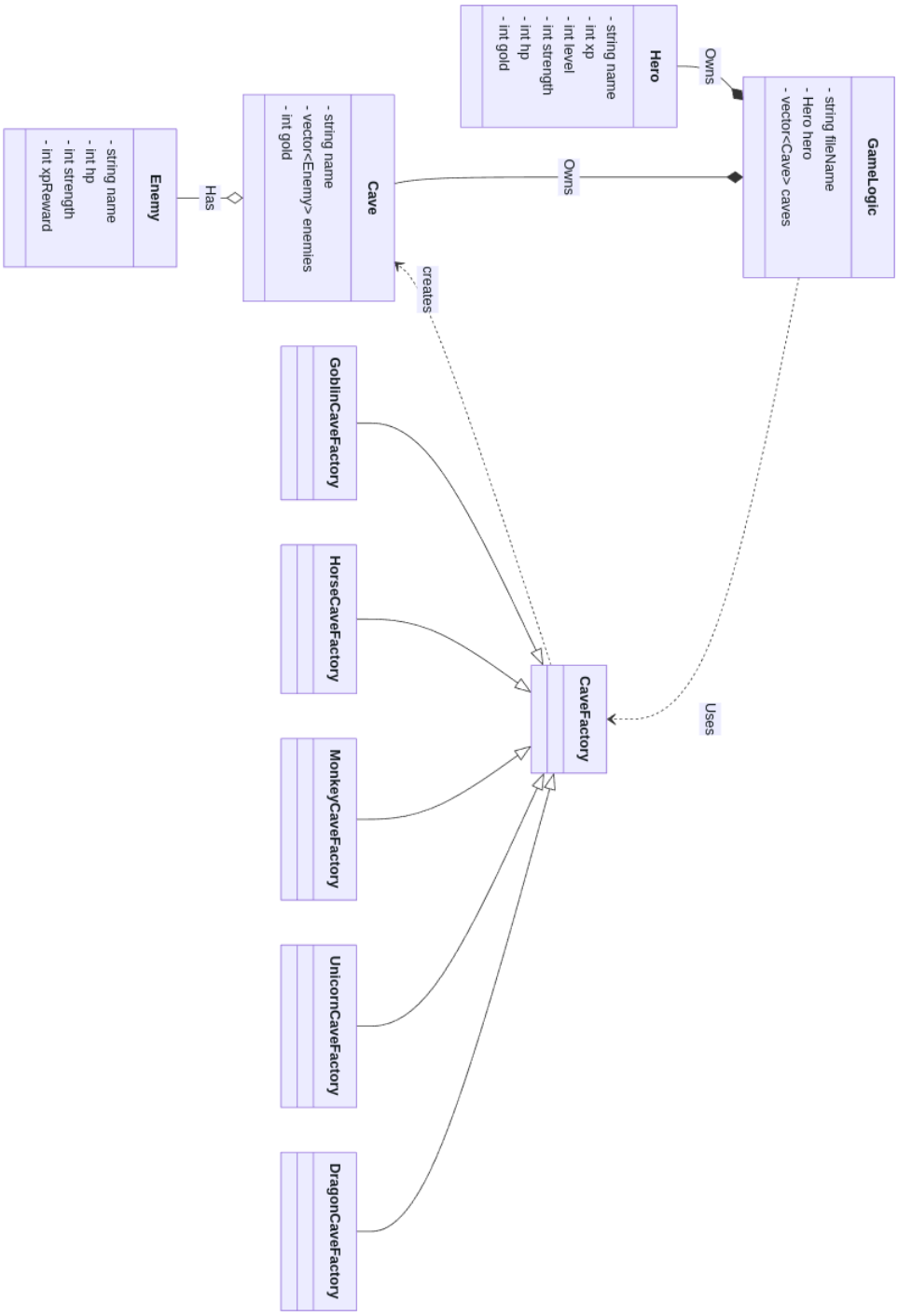

Figure 7: Domain model

2.4 Sekvens diagramer

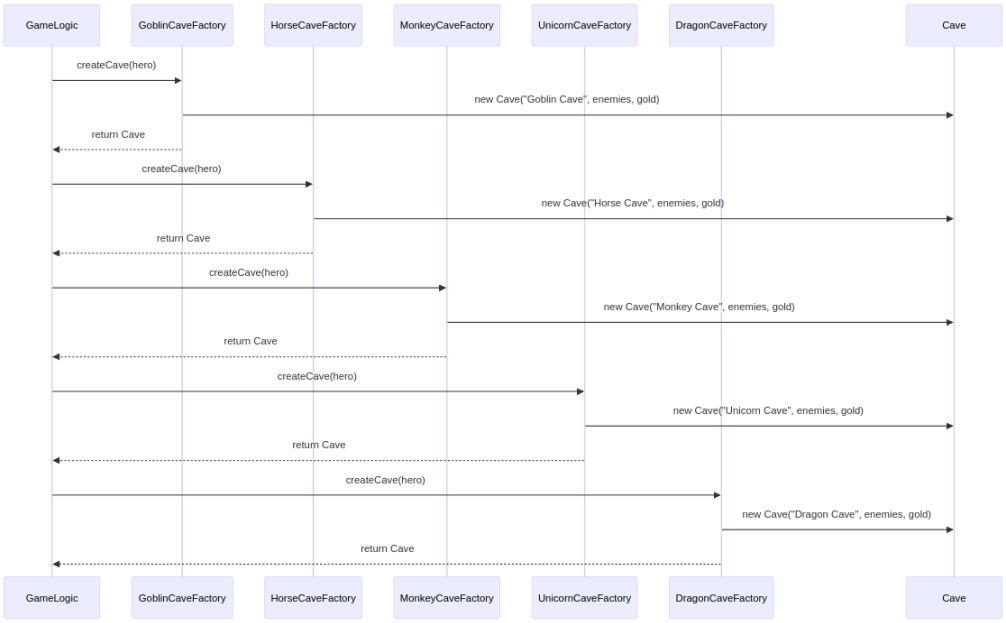

Figure 8: Sekvens diagaram

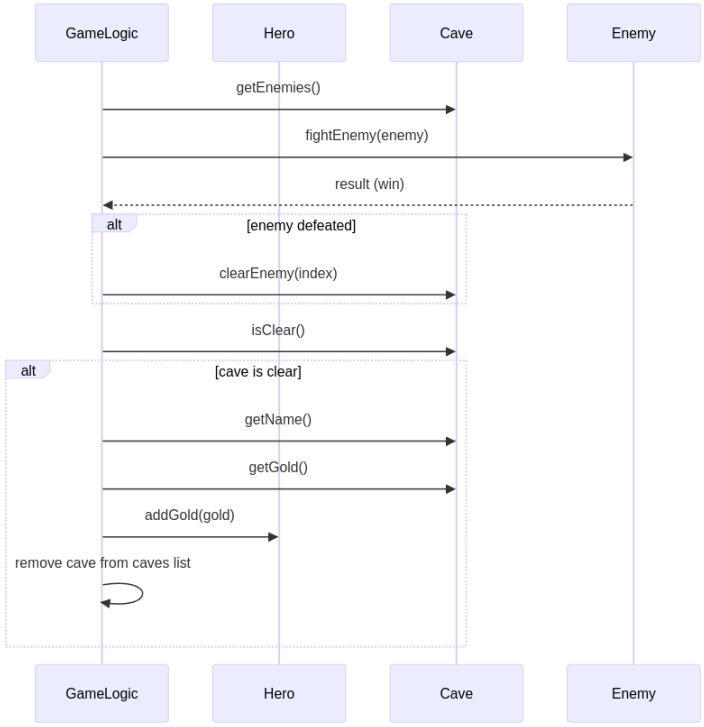

Figure 9: Sekvens diagram

2.5 UML class diagram

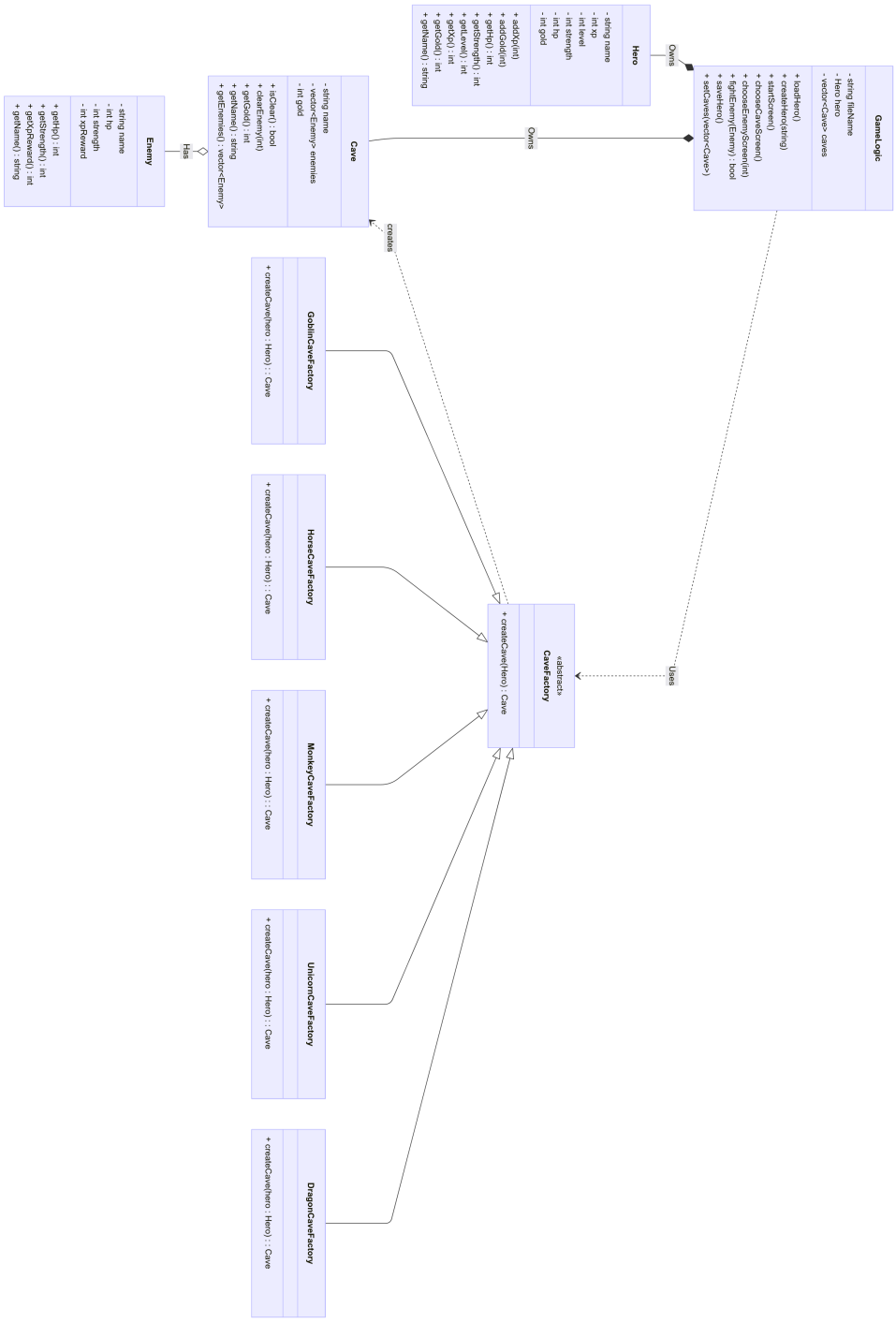

Figure 10: UML class diagram

3 Iteration

Det er nu muligt at købe våben til hero for det gold som er optjent ved at gennemføre grotter. Våben giver ekstra strength samt en multiplier og har en durablity og går i stykker efter brug. Der ud over er alt data for gemte heros blevet flyttet fra en .txt fil til en .db database. Som kan blive læst med SQL. Stats for spillet bliver også gemt. Så det er muligt at kunne se statestikker for heros der er gemt. Alle kills bliver gemt , samt hvilken hero og våben der udførte kill. Så det kan ses hvormange kills hver hero har, og med hvilke våben.

3.1 Use cases

- **Køb våvben** - Når spiller er i menuen kan spiller åbne amory, hvor spiller kan købe forskellige våen med "gold" optjent fra grotter. Våben skal have en skade der bliver lagt til hero styrke, en styrk modifier som bliver gnaget på total hero styrke, og en holdbarhed der falder ved brug. Våben bliver tilføjet til hero.
- **Brug af våben** - Hvis hero ahr et våben og spiller vælger at kæmpe mod en fjende vil våbenet blive brugt. For hvert slag i kampen mister våben 1 holdbarhed. Hvis våbenets holdbarhed = 0 gå det i stykker of fjernes fra hero.
- **Gem hero** - Spiller skal kunne gemme hero til en database. Her gemmes alt information om hero, hp, styrke, xp, lvel, våben, navn. Så hero kan loades af spiller neste gnag sil startes.
- **Load hero** - Når spillet startes skal spiller have mulighed for at load hero fra database. Heros skal vises i alfabetisk rekkefølge, og spiller vælger hvilken at spille som så fremt der er heros i databasen.
- **Vis stats** - Spiller skal fra menuen kunne vælge at se spil statestikker. Her skal spiller kunne se stats fra alle heros. Heros vises i alfabetisk rækkefølge. Det skal fremgå hvor mange monstre hero har besejret.
- **Våben kills** - Spiller skal kunne se statestik for hvor mange kills en hero har med hvert våben. Spiller skal også kunne se hvilken Hero der har besejret flest fjender med et givet våben.

3.2 Use case diagram

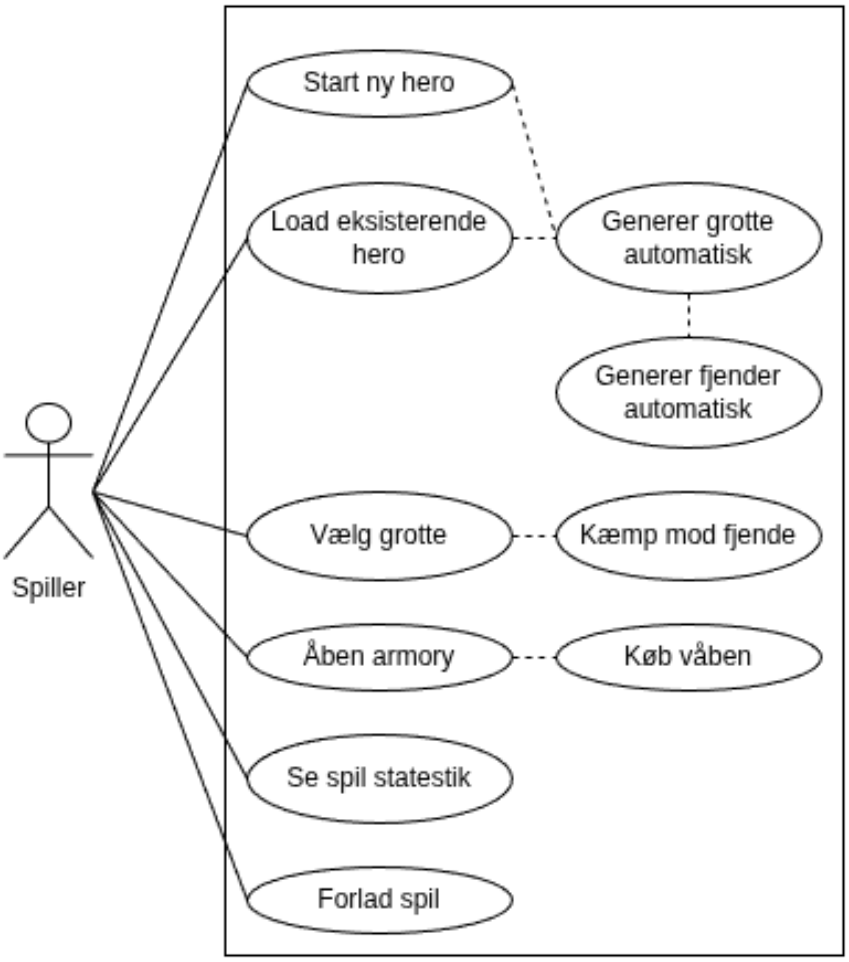

Figure 11: Use case diagram

3.3 Domain model

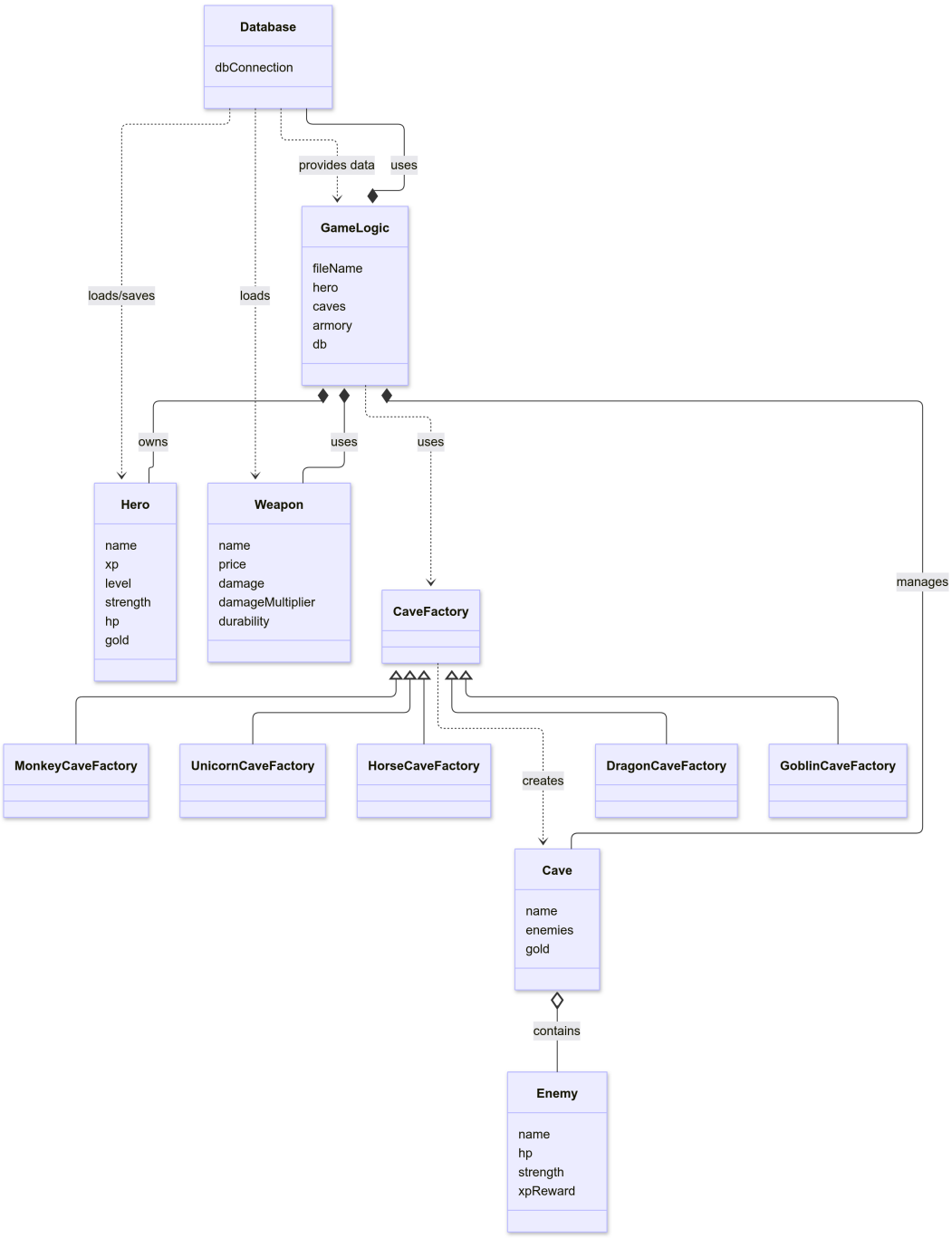

Figure 12: Domain model

3.4 Sekvens diagramer

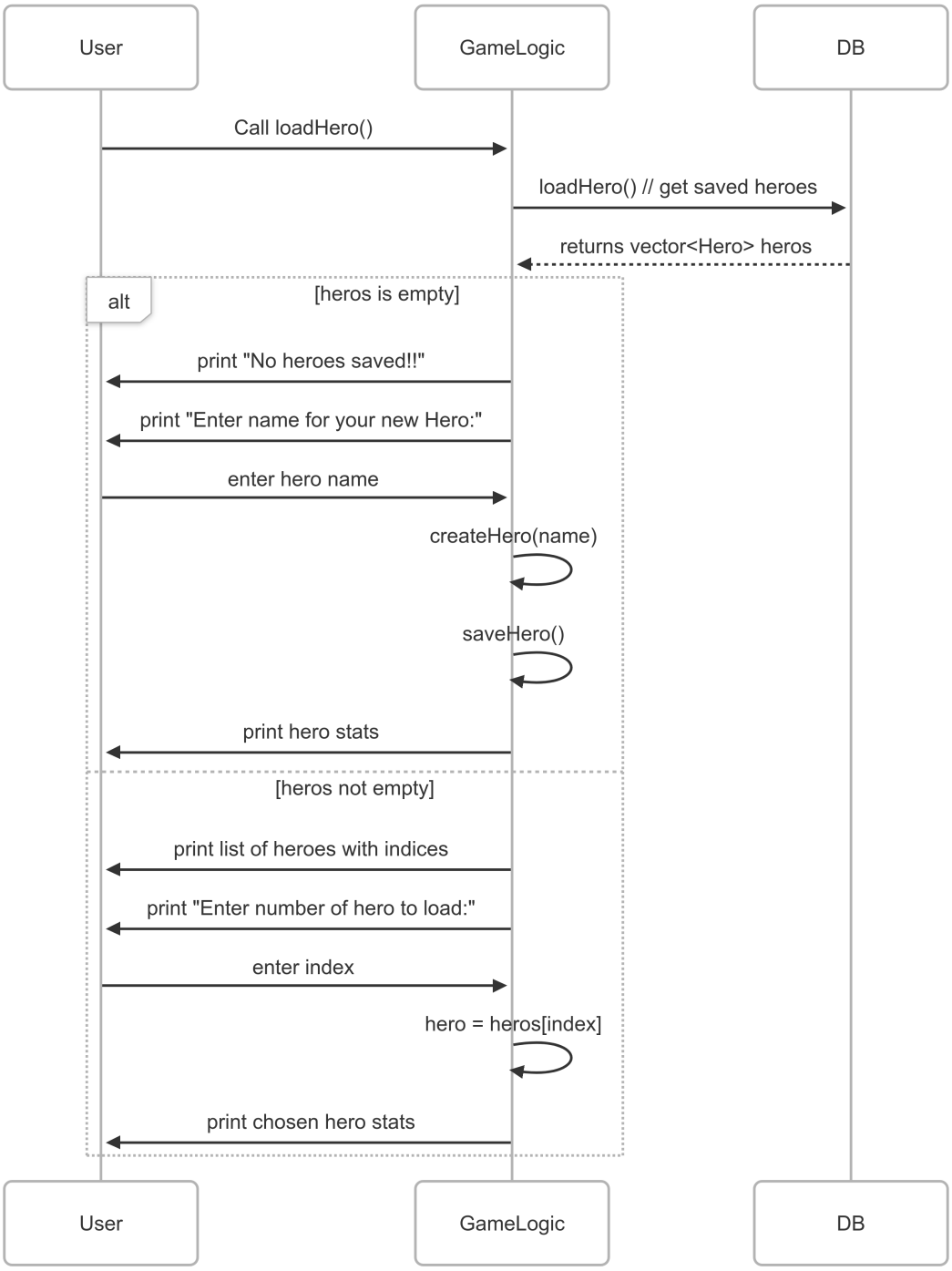

Figure 13: Sekvens diagaram

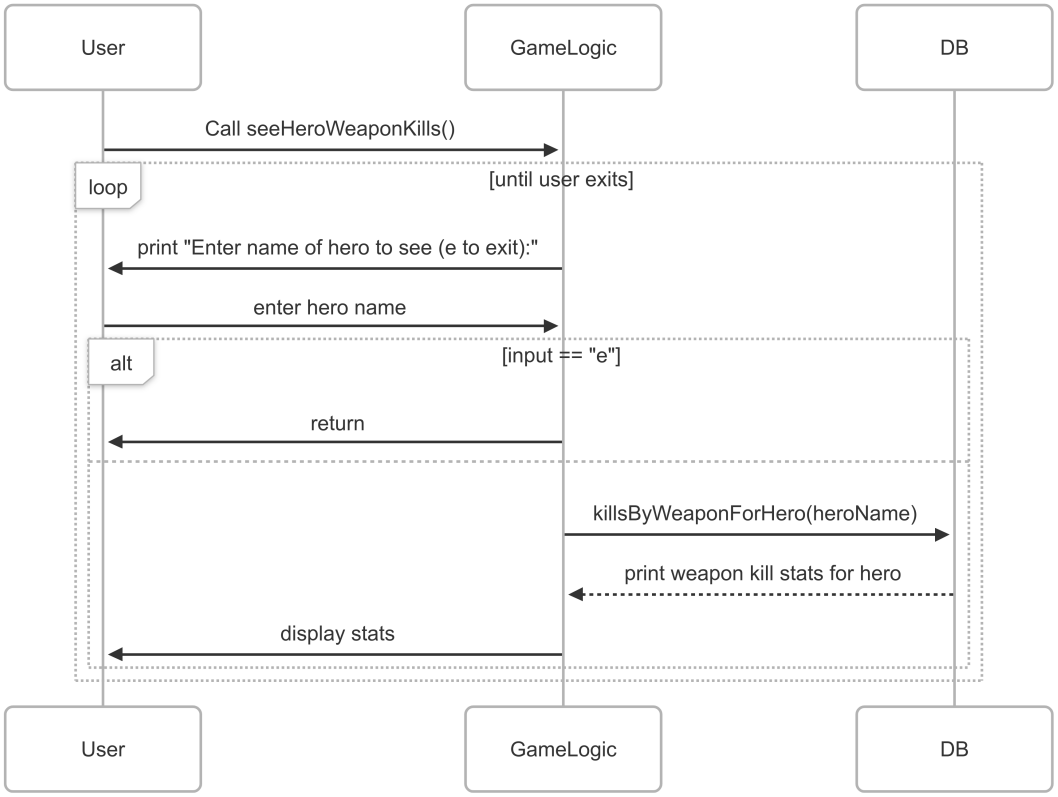

Figure 14: Sekvens diagram

3.5 UML class diagram

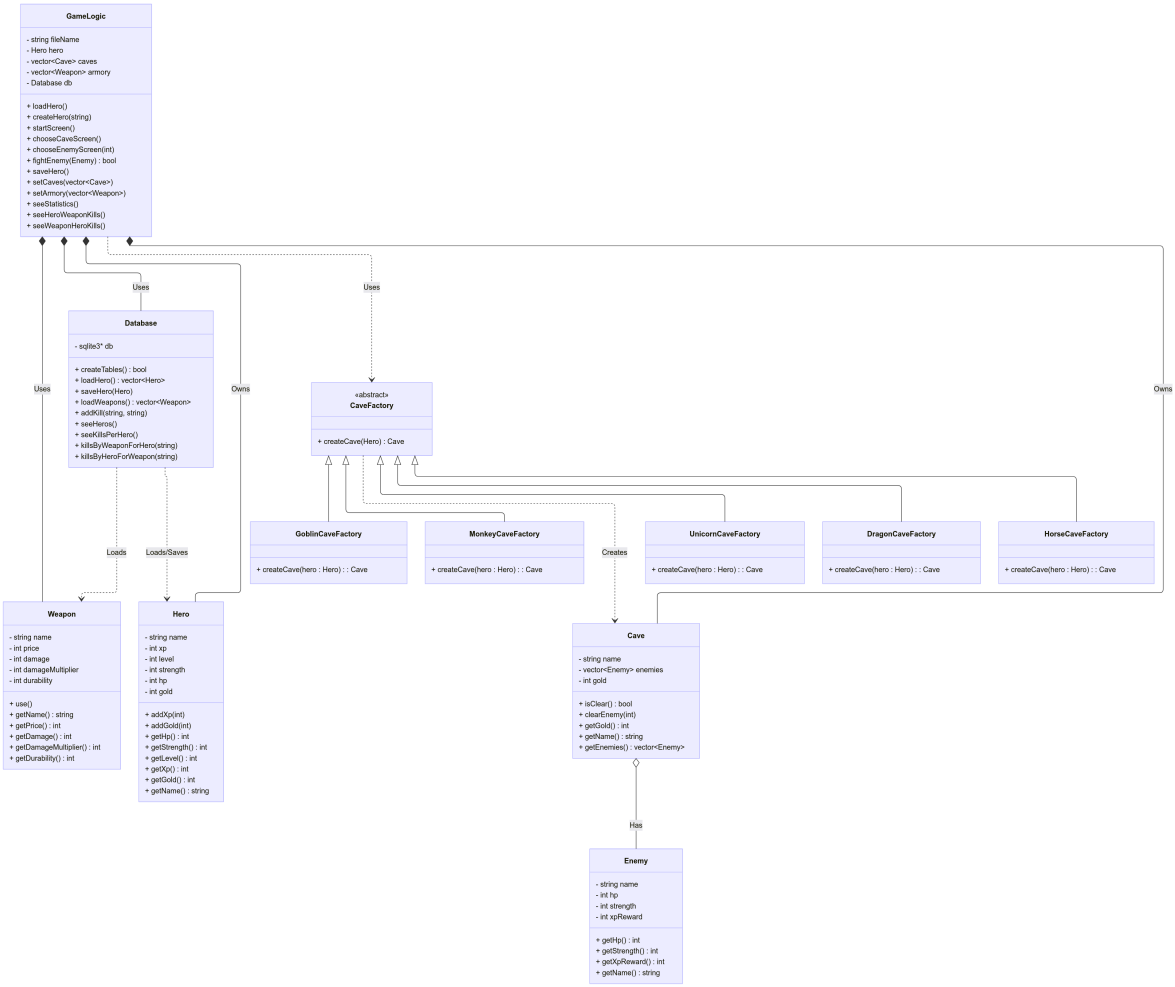

Figure 15: UML class diagram