

nWOD Hahmotietokanta

1 Johdanto

Järjestelmä on tarkoitettu pelinjohtajan avuksi pelatessa New World of Darkness pöytäroolipeliä. Sen avulla pelinjohtajan tulisi saada helposti katsottua hahmojen tietoja pelin aikana joutumatta kysymään niitä pelaajilta tai katsomaan niitä pelaajien hahmopapereista. Tämän tulisi tehdä pelaamista sujuvammaksi ja estää pelaajia saamasta vihjeitä pelistä heiltä kysyttäessä.

Tavoitteena olisi, että pelinjohtaja saisi muutamalla klikkauksella löydettyä minkä tahansa haluamansa tiedon (esim. yhden hahmon statit tai kaikkien hahmojen tietyn taidon arvo).

Pelinjohtajan tulisi myös saada lisättyä, poistettua ja muokattua kaikkien taulukoiden tietoalkioita.

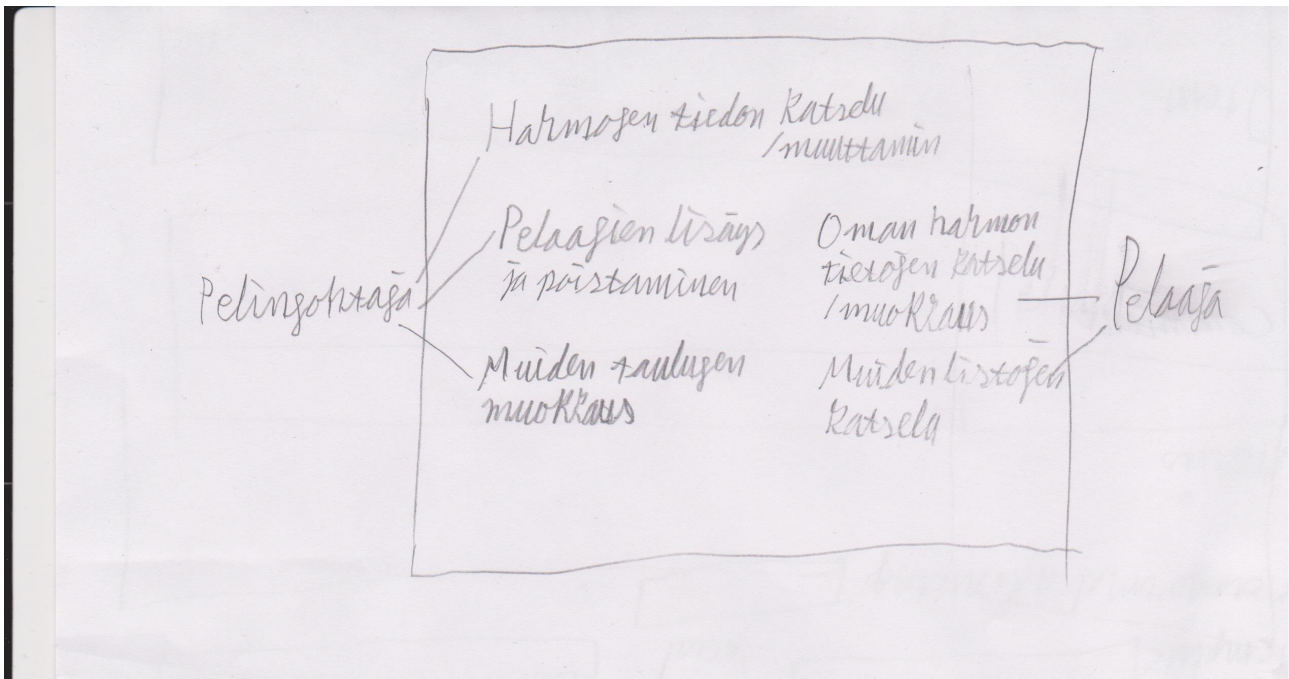
Järjestelmän ohjelmakoodi on toteutettu Javalla ja tietokanta PostgreSQL:ällä. Serverinä käytetään Apachen Tomcat-serveriä. Täten sovelluksen alustajärjestelmän on tuettava Javaa ja sille on oltava asennettuna Tomcat-serveri. Sovellus vaatii myös PostgreSQL-tukea sen alustajärjestelmästä, vaikka jokin muukin SQL saattaisi myös toimia. Postgresia kuitenkin suositellaan.

2 Yleiskuva

Järjestelmässä kuului olla kaksi käyttäjäryhmää, pelinjohtaja ja pelaajat. Mutta aikarajoitteiden takia käyttäjäryhmien oikeuksia ei ehditty implementoida, joten kaikilla käyttäjillä on täydet oikeudet järjestelmässä.

Pelinjohtajalla kuului olla kaikki mahdolliset oikeudet järjestelmässä, eli taulukoiden vapaa lukeminen ja muokkaaminen. Pelinjohtajan vastuulla olisi ollut uusien tietoalkioiden luonnin ja vanhojen muokkaamisen. Tähän olisi sisällynyt uusien pelaajien lisäämisen.

Pelaajien oikeudet olisivat olleet paljon rajoitetummat. Pelaajilla olisi ollut oikeudet vain omien hahmojensa tarkisteluun ja niiden tietojen muuttamiseen sekä muiden taulujen sisällön katseluun. Täten pelaajat olisivat voineet tarkastella olemassa olevia merittejä, taitoja tai esineitä jotta heillä olisi ollut mahdollisuus lisätä niitä omalle hahmolleen.



Yllä käyttötapauskaavio

3 Käyttötapaukset

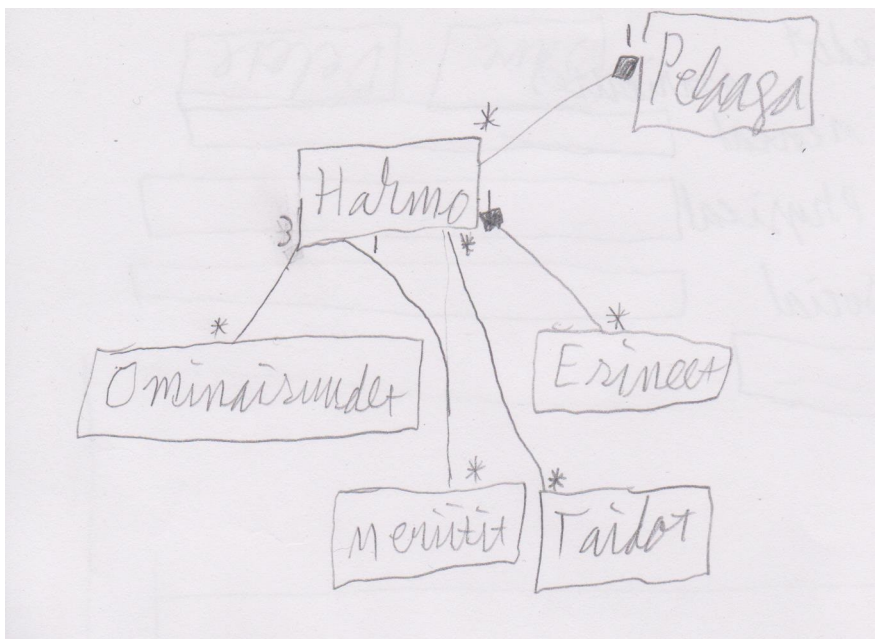
Pelaajan käyttötapaukset:

- Omien hahmojen tietojen lukeminen: Pelaaja kykenee katsomaan vapaasti jokaisen hahmonsa tietoja.
- Olemassa olevien meriittien, taitojen ja esineiden katselu: Pelaaja kykenee katsomaan olemassa olevia meriittejä, esineitä ja taitoja, muttei kykene näkemään mitä edellämainittuja asioita muiden pelaajien hahmoilla on.
- Omien hahmojen tietojen muuttaminen: Pelaaja kykenee muuttamaan hahmonsa kenttien tietoa vapaasti.

Pelinjohtajan käyttötapaukset:

- Tietoalkioiden luonti: Pelinjohtaja kykenee luomaan uusia meriittejä, esineitä, taitoja hahmoja ja pelaajia. Pelaajaa luodessa lisätään uusi käyttäjä järjestelmään, joten pelinjohtaja on vastuussa muiden käyttäjien käyttäjänimistä ja salasanoista.

4 Järjestelmän tietosisältö



Käsitekaavio

Attribuutit	Arvojoukko	Kuvailu
Name	Merkkijono, max 30 merkkiä	Hahmon nimi
Player	Merkkijono, max 30 merkkiä	Hahmon pelaajan nimi
Mental	Merkkijono, max 10 merkkiä	Hahmon älyllisten ominaisuuksien (intelligence/wits/resolve) arvot
Physical	Merkkijono, max 10 merkkiä	Hahmon fyysisten ominaisuuksien (strength/dexterity/stamina) arvot
Social	Merkkijono, max 10 merkkiä	Hahmon sosiaalisten ominaisuuksien (presence/manipulation/composure) arvot.
Template	Merkkijono, max 10 merkkiä	Jos hahmo on jokin muu kuin tavallinen kuolevainen, merkitään se tähän.
Metanormal advantage	Integer 0-10	Hahmon templateen liittyvän yliluonnollisen voiman arvo. Jos hahmo on ihminen arvo on 0.

Tietokohde: Hahmo

Pelissä pelaajat eläytyvät kerrallaan yhden hahmon rooliin ja koittavat toimia niinkuin hahmot toimisat pelissä eteen tulevissa tilanteissa. Hahmot ovat uniikkeja ja jokaisella on oma persoonallisuutensa, taustansa, ominaisuutensa ja taitonsa. Tähän järjestelmään tallennetaan vain ominaisuudet, taidot, meriitit ja esineet. Taustat ja muut jätetään pelaajien ja pelinjohtajan muistettavaksi.

Attribuutit	Arvojoukko	Kuvailu
Name	Merkkijono, max 30 merkkiä	Pelaajan nimi. Sitä käytetään myös järjestelmään sisäänkirjautumiseen
ID	Merkkijono, max 30 merkkiä	Turhaksi jäänyt attribuutti, jota ei ole ehditty poistaa sovelluksesta.
Passwd	Merkkijono, max 30 merkkiä	Pelaajan salasana jolla hän kirjautuu järjestelmään.

Tietokohde: Pelaaja

Pelaajat ovat järjestelmän käyttäjiä. Alunperin yhden pelaajan kuului olla pelinjohtaja jolla olisi ollut muita korkeammat oikeudet järjestelmässä, mutta aikarajoitteiden takia kaikilla käyttäjillä on täydet oikeudet. Yhdellä pelaajalla voi olla useita hahmoja, mutta yhdellä hahmolla voi olla vain yksi pelaaja.

Attribuutit	Arvojoukko	Kuvailu
Value	Merkkijono, max 10 merkkiä	Ominaisuuden arvo
Character	Merkkijono, max 30 merkkiä	Hahmon nimi

Tietokohde: Ominaisuudet

Ominaisuus taulua ei ehditty implementoimaan järjestelmään. Ideana oli, että taulussa olisi ollut vain kolme riviä, yksi jokaiselle ominaisuus tyypille. Ja hahmojen arvot olisivat näkyneet taulun sarakkeina.

Attribuutit	Arvojoukko	Kuvailu
Name	Merkkijono, max 30 merkkiä	Esineen nimi
Character	Merkkijono, max 30 merkkiä	Esineen omistajan nimi

Tietokohde: Esineet

Esineet ovat hahmojen omistamia tavaroita. Pelissä esineillä ei ole niin kovin suurta merkitystä. Elleivät ne ole erityisen harvinaisia/tärkeitä niitä ei edes merkitä ylös. Tämän takia tietokantaan kirjatut esineet ovat uniikkeja ja niillä on uniikkit nimet. Siinä tapauksessa että kahdella tai useammalla hahmolla on identtisiä esineitä joudutaan ne nimeämään tietokantaan eritavalla.

Attribuutit	Arvojoukko	Kuvailu
Name	Merkkijono, max 30 merkkiä	Taidon nimi
Character	Merkkijono, max 30 merkkiä	Taidon omaavan hahmon nimi
Value	Integer 0-10	Taidon arvo, eli kuinka harjaantunut kyseinen hahmo on tässä taidossa.

Tietokohde: Taidot

Taitoja ei ehditty implementoida. Tarkoituksena oli luoda taitojen ja hahmojen välille monesta moneen suhde, koska taitoja on rajallinen määrä ja hahmoilla on jokaisen taidon arvo vähintään 0.

Attribuutit	Arvojoukko	Kuvailu
Name	Merkkijono, max 30 merkkiä	Meriitin nimi
Character	Merkkijono, max 30 merkkiä	Meritin omaavan hahmon nimi
Value	Integer 1-10	Meriitin arvo

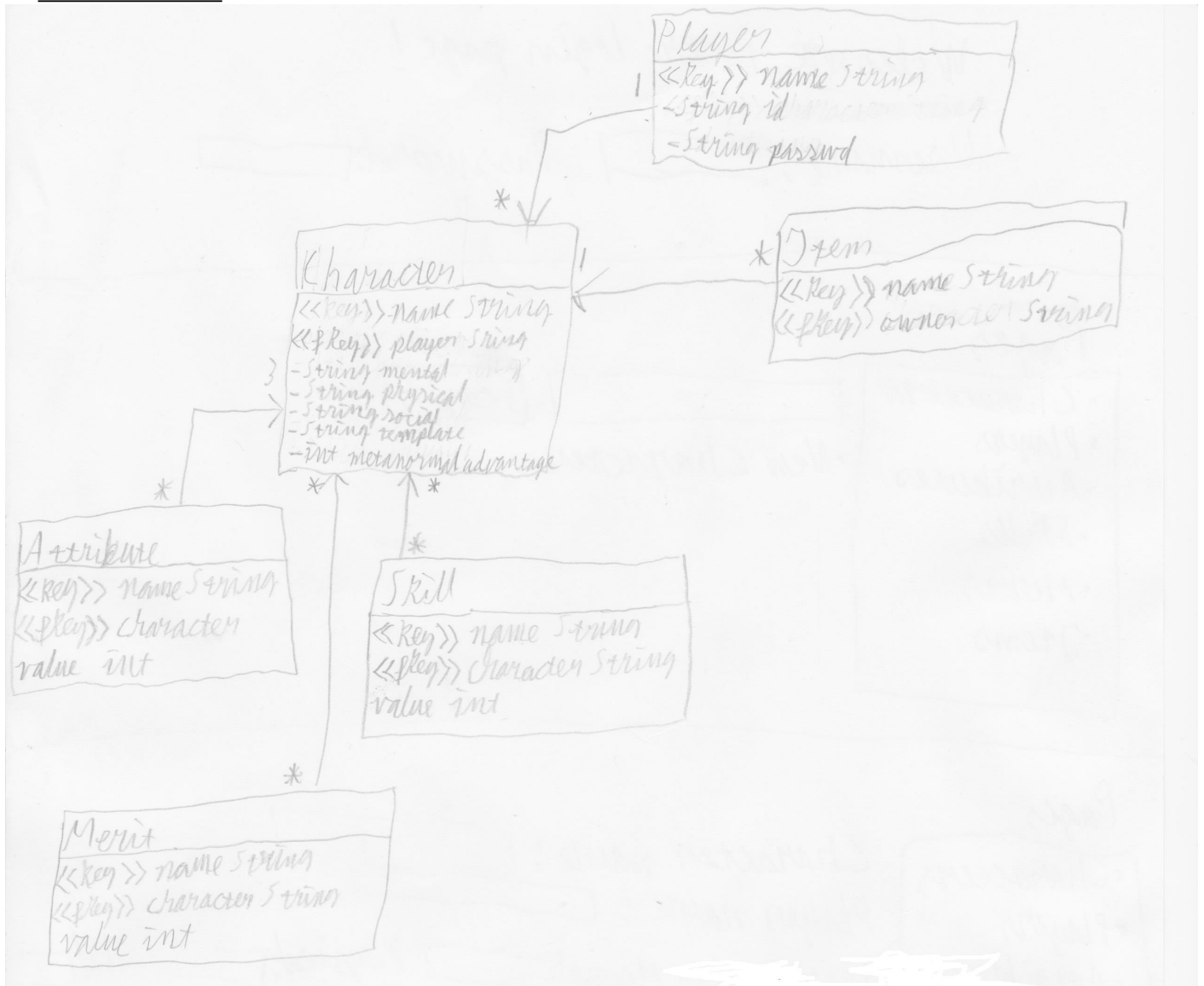
Tietokohde: Meriitit

Pelissä meriitit ovat jotain asioita jotka vaikuttavat hahmoon merkittävällä tavalla mutta eivät ole kykyjä taitoja tai esineitä. Ne voivat olla lähes mitä tahansa. Merittejä ei aikarajoitteiden takia ehditty implementoida. Taitojen tapaan niillä ja hahmoilla piti olla monesta moneen suhde, koska on hyvin todennäköistä että hahmoilla olisi samoja meriittejä.

-Tietoalkioiden poistaminen: Pelinjohtaja kykenee poistamaan edellämainittuja tietoalkioita.

-Taulujen vapaa katselu: Pelinjohtaja kykenee katselemaan kaikkien taulun kaikkia tietoja vapaasti.

5 Relaatiokaavio



Create ja drop table tiedostot löytyvät kansiossa TietokantaSov/src/java/tietokanta.

6 Järjestelmän yleisrakenne

Projektissa on käytetty karsittua MVC-mallia. Kontrollereita ei ole, vaan niiden virasta huolehtivat servletit. Viewejä ei myöskään tarkalleen ottaen ole ja niiden virkaa puolestaan hoitavat jsp sivut. Modelit ovat samalla tietokantaan talletettavia olioita sekä niiden toiminasta vastaavia luokkia. Modelien metodit on jaettu staattisiin ja ei staattisiin metodeihin. Tietokantaan vieminen hoidetaan luomalla tarvittava olio ja käyttämällä olion omaa metodia. Tietokannasta hakeminen ja tietokohteen poistaminen puolestaan hoidetaan staattisella metodilla. Modelit sijaitsevat kansiossa sourcehahmolista/models

Servletit ovat rajapinta jsp-sivujen ja toimintalogiikan välillä. Niiden kautta tieto kuljetetaan tietokannasta sivulle ja sivulta tietokantaan. Myös virheellisteiden syötteiden tarkistaminen hoidetaan servleteissä. Servletit sijaitsevat kansiossa hahmolista/servlets.

Jsp sivut ovat vastuussa käyttöliittymästä ja tietokannasta haetun tiedon esittämisestä. Ne sijaitsevat web-kansiossa. Webissä ovat myös servlettien mappamiseen käytetty web.xml ja bootstrap-tiedostot. Web.xml on kansiossa web/web-inf ja bootstrapit löytyvät kansioista web/css

Tietokannan create- ja drop-table tiedosto sijaitsevat tietokanta-kansiossa. Apukirjastot

sijaitsevat Libraries kansiossa.

7 Järjestelmän komponentit

Jsp sivut:

Login.jsp on sovelluksen etusivu jonka kautta kirjaudutaan sivustolle. Onnistuneen sisäänkirjautumisen jälkeen se ohjaa käyttäjän character.jsp:lle.

Character.jsp tällä sivulla löytyvät tietokannan hahmot. Sieltä löytyvät myös napit tietyn hahmon hakemiselle ja hahmon poistamiselle. Napit käyttävät samaa input kenttää, joten haettaessa/poistaessa kannattaa olla varovainen. Viimeiseksi siellä on linkki newCharacter-sivulle jossa voi lisätä hahmon tietokantaan.

NewCharacter.jsp:tä löytyvät kentät hahmon tietojen syöttämiseksi. Siellä on kentät hahmon nimelle, pelaajan nimelle, mental-, physical- ja social attributeille, templatelle ja metanormal advantagelle.

Player.jsp on pelaajille sama kuin character.jsp on hahmoille. Perusnäkyssä sinne on tulostettu kaikkien pelaajien nimet ja sieltä löytyy character.jsp:n tapaan tekstikenttä johon kirjoitetaan pelaajan nimi ja sen vieressä on napit joilla kyseinen pelaaja etsitään tietokannasta tai poistetaan sieltä. Alimpana löytyy linkki newPlayer sivulle jossa luodaan uusi pelaaja.

NewPlayer.jsp on vastuussa uusien pelaajien syöttämisestä tietokantaan. Siellä on kentät pelaajan nimelle, id:lle ja salasalle.

Item.jsp on samanlainen kuin character.jsp ja player.jsp. Sieltä löytyvät tietokantaan tallennetut esineet ja sieltä on mahdollista etsiä ja poistaa esineitä. Siellä on myös linkki newItem sivulle jossa syötetään uusia esineitä tietokantaan.

NewItem.jsp täällä syötetään uusia item-olioita tietokantaan. Täältä löytyy kentät esineen nimelle ja sen omistajan nimelle.

Tagi.tag on.jsp sivujen käyttämä tiedosto josta löytyy järjestelmän sivukartta.

Models luokat:

Character on vastuussa character-olioista sekä niiden tallettamisesta, poistamisesta ja etsimisestä tietokannasta. Se on jaettu staattisiin ja ei staattisiin luokkiin, joista staattiset ovat vastuussa olioiden etsimisestä ja poistamisesta ja ei staattiset olioiden luomisesta ja niiden tallentamisesta tietokantaan.

Player on vastuussa player-olioista. Ne kuvaavat järjestelmän käyttäjiä ja niihin on tallennettu pelaajan nimi, pelaajan id (jota ei koodissa käytetä mihinkään) ja pelaajan salasana. Luokka on muiden models luokkien tapaan jaettu staattisiin ja ei staattisiin metodeihin, joista staattiset ovat vastuussa etsimisestä ja poistamisesta, ja ei staattiset tietokantaan tallentamisesta.

Item on vastuussa item-olioista. Sinne on tallennettu esineen nimi sekä sen omistajan nimi. Luokka on muiden models luokkien tapaan jaettu staattisiin ja ei staattisiin metodeihin, joista staattiset ovat vastuussa etsimisestä ja poistamisesta, ja ei staattiset tietokantaan tallentamisesta.

Modellsista löytyvät myös luokat: AttributeList, MeritList ja SkillList mutta ne ovat kaikki tyhjiä eikä niitä läydetä mihinkään koska niitä ei ehditty tehdä.

Servlet luokat:

CharacterServlet on vastuussa Character luokan ja character.jsp:n välisestä kommunikaatiosta. Kuuntelee osoitetta /char.

PlayerServlet toimii Player luokan ja player.jsp:n välisenä rajapintana. Se välittää player.jsp:ltä tulevat käskyt ja kutsuu tarvittavia Player luokan metodeja. Kuuntelee osoitetta /player.

ItemServlet toimii Item luokan ja item.jsp:n välisenä rajapintana. Se välittää item.jsp:ltä tulevat käskyt ja kutsuu tarvittavia Item luokan metodeja. Se kuuntelee osoitetta /item.

NewCharacterServlet on vastuussa newCharacter.jspn palvelemisesta. Se kuuntelee osoitetta /NewCharacter ja kutsuu tarvittavia Character luokan metodeja. Se myös hoitaa syötteiden tarkistamisen.

NewPlayerServlet on vastuussa newPlayer.jspn palvelemisesta. Se kuuntelee osoitetta /NewPlayer ja kutsuu tarvittavia Player luokan metodeja. Se myös hoitaa syötteiden tarkistamisen.

NewItemServlet on vastuussa newItem.jspn palvelemisesta. Se kuuntelee osoitetta /NewItem ja kutsuu tarvittavia Item luokan metodeja. Se myös hoitaa syötteiden tarkistamisen.

Login hoitaa järjestelmään kirjautumisen. Se kuuntelee osoitetta /login ja kutsuu Playerin haePelaaja-metodia käyttäjätunnuksen ja salasanan tarkistamiseen. Login myös hoitaa istunnon tallentamisen.

Logout hoitaa järjestelmästä uloskirjautumisen. Se kuuntelee osoitetta /logout ja sen ainoa tehtävä on poistaa sessiosta kirjautunut-attribuutti ja palauttaa käyttäjä sen jälkeen etusivulle.

TemplateServlet on kaikkien muiden servlet luokkien paitsi loginin ja logoutin perimä luokka. Se hoitaa kirjautumisen tarkistamisen sivua avattaessa.

8 Käyttöliittymä

Sivut käyttävät sivupalkkia, mikä tarkoittaa että kaikilta sivuilta pääsee Character, Player ja Item sivuille. Mutta vain pääsivuilta pääsee sivulle missä luodaan uusi tietoalkio. Esim vain Character sivulta pääsee newCharacter sivulle missä luodaan uusi hahmo ja viedään se tietokantaan. Miltä tahansa sivulta on myös mahdollista kirjautua ulos. Muita sivuja paitsi kirjautumissivua pääsevät käyttämään vain kirjautuneet käyttäjät.

alkioiden tulostaminen, yhden alkion etsiminen, alkion luominen ja alkion poistaminen.

Tunnettuja bugeja ovat kirjautuessa jos jättää käyttäjänimen tyhjäksi tai kirjoittaa väärän käyttäjänimen sivu ei esitä virheilmoitusta vain menee valkoiseksi. Uusia tietoalkioita luodessa syötteen tarkistus ei myöskään osaa tarkistaa onko kyseistä primary keytä jo käytetty vaan heittää virhesivun jos koittaa tallettaa tietokantaan alkion jo käytetyllä primary keyllä.

Ohjelmasta puuttuu kokonaan mahdollisuus muokata jo olemassa olevia tietoalkioita, joten mahdollinen muokkaaminen on hoidettava poistamalla vanha alkio ja luotava päivitetty versio sen tilalle. Myöskään luokkia merit, attribute ja skill ei ole ollenkaan olemassa ja niiden sivuille meneminen aiheuttaa virheen.

12 Omat kokemukset

Kurssi oli hyvin työläs ja ohjeet olivat välillä epäselviä tai vajavaisia. Kurssisivula olisi hyvä olla linkkejä tutoriaaleihin tai muuhun vastaavaan mistä voisi hakea tietoa. Nyt internetin koluaminen jonkun ongelmatilanteen sattuessa oli aikaavievä eikä silloin yleensä ymmärtänyt kunnolla miten löytämä asia oikeasti toimii.

13 Liitteet

-Ohjelmakoodi ja create tablet löytyvät osoitteesta <https://github.com/Hattumies/Tietokantasovellus>.

-Ohjelmakoodi:

<https://github.com/Hattumies/Tietokantasovellus/tree/master/TietokantaSov/src/java/hahmolista>

-Create tablet:

<https://github.com/Hattumies/Tietokantasovellus/tree/master/TietokantaSov/src/java/tietokanta>

-Jsp sivut: <https://github.com/Hattumies/Tietokantasovellus/tree/master/TietokantaSov/web>