

# DnD3.5 Character Builder

Henri Vehmas  
014024454

## Sisällysluettelo

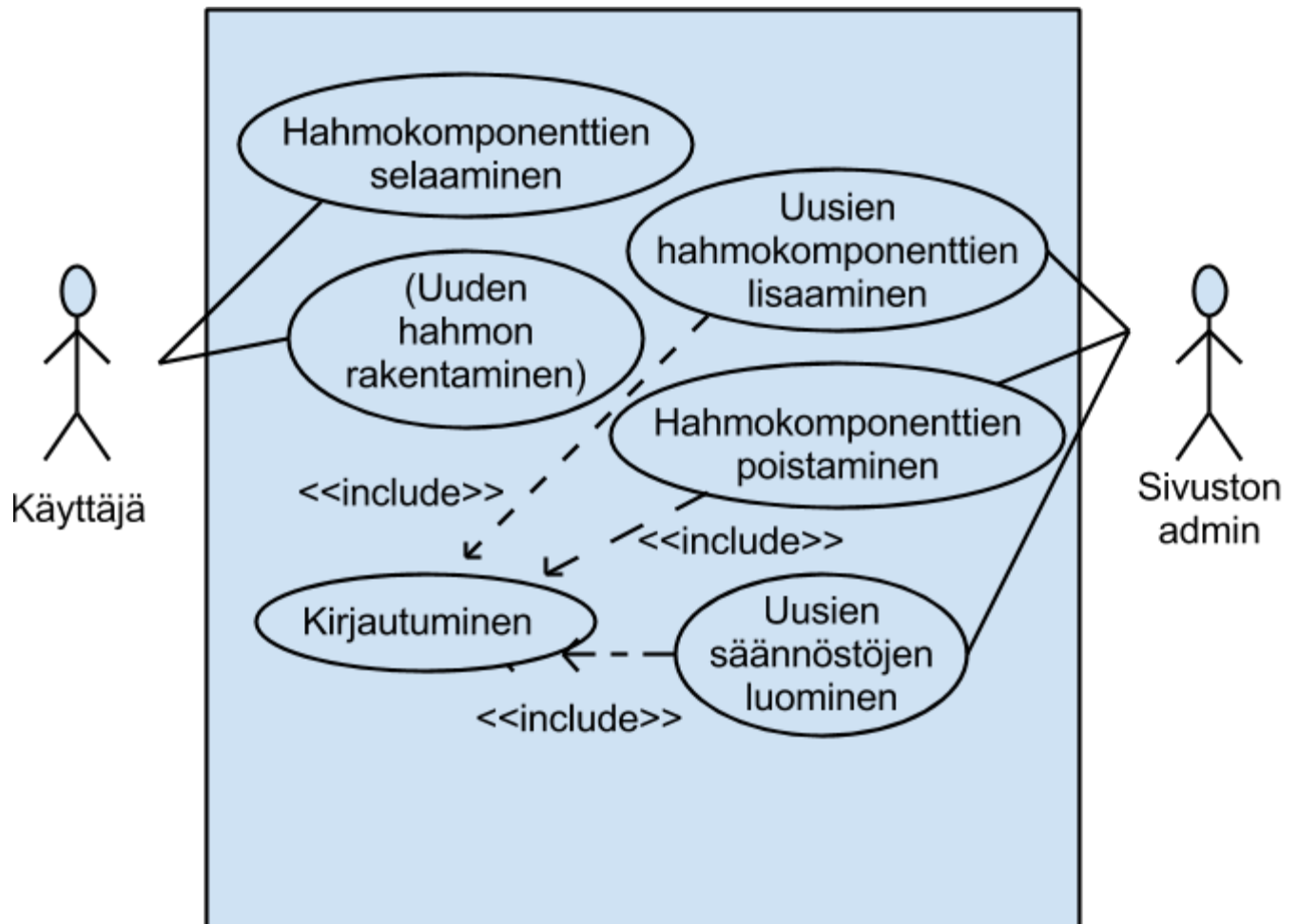
- [3--Johdanto](#)
- [4--Käyttötapauskaaviot](#)
  - [5--Käyttäjäryhmät](#)
  - [6--Käyttötapaukset](#)
    - [6--Käyttäjän käyttötapaukset](#)
    - [6--Sivuston adminin käyttötapaukset](#)
- [7--Järjestelmän tietosisältö](#)
  - [10--Tietokohteiden tietosisältöjen kuvaukset](#)
- [14--Järjestelmän yleisrakenne](#)
- [15--Järjestelmän komponentit](#)
  - [15--JSP-sivut:](#)
  - [21--CSS:](#)
  - [21--Servletit:](#)
  - [22--Apuluokat:](#)
  - [22--Entityluokat:](#)
- [23--Käyttöliittymä](#)
- [24--Asennustiedot](#)
- [25--Käynnistys-/Käyttöohje](#)

## Johdanto

Järjestelmän tarkoituksena on mahdollistaa Dungeon & Dragons 3.5 säännösten hahmonluontiin liittyvien komponenttien ja niiden yhteyksien tehokas luominen ja varastoiminen. Lopullisena tavoitteena on hahmon rakentaminen alusta loppuun lomakkeellisesti, sekä hahmon tietojen tallentaminen tulostettavissa olevaan järjestettyyn tekstimuotoon, tämä ominaisuus on kuitenkin tarkoitus toteuttaa pidemmällä aikavälillä (siis toiminnallisuus ei ole toteutettuna vielä tsoha-kurssilla).

Itse ohjelman toteutus on java-pohjainen, ja tietokantapuoli on hoidettu PostgreSQL:llä. Tarvittaessa ohjelma on kuitenkin mahdollista siirtää jollekin toiselle tietokantapohjalle suhteellisen helposti. Sivusto ei tällä hetkellä käytä yhtään javascriptiä.

## Käyttötapauskaaviot



Selkeyden vuoksi kaaviossa ei ole esitetty "hahmokomponenttien" sisältöä tarkemmin. Hahmokomponentit, hahmon ominaisuuksia määrittävät elementit, on tässä tapauksessa jaettu suunnilleen seuraavasti:

- Hahmon rotu, Race
- Hahmoluokka, Class
- Hahmon taidot, Skills
- Hahmon erikoistaidot, kyvyt yms., Feat
- Hahmon palvoma jumaluus, Deity
- Jumaluuksien valta-alueet, Domain

Classeihin ja Featteihin kohdistuu valtaisa määrä erilaisia vaatimuksia, jotka käytännössä muodostavat hahmonluomisen ytimen.

## Käyttäjäryhmät

### Käyttäjä

- Kuka tahansa sivustoa käyttävä henkilö.

### Sivuston admin

- Kaikesta vastuussa oleva henkilö, joka voi sivuston kautta lisätä sekä kokonaan uusia säännöstöjä että yksittäisiä komponentteja, muokata yhteyksiä ja poistaa mitä haluaa.

## Käyttötapaukset

### Käyttäjän käyttötapaukset

Hahmokomponenttien selaaminen:

Käyttäjä voi tarkastella joko kaikkia johonkin kategoriaan kuuluvia komponentteja, tai hakea niitä niiden nimen perusteella.

\*Hahmon rakentaminen:

Käyttäjä voi vaiheittain etenevällä järjestelmällä rakentaa uuden hahmon käyttäen niitä komponentteja, minkä vaatimukset hahmo milloinkin täyttää. Kun käyttäjä on saanut hahmon haluamaansa vaiheeseen, voi hän sitten tallentaa hahmon tiedot erillisenä tiedostona helposti tulostettavissa olevassa muodossa. Ei vielä toteutettu.

### Sivuston adminin käyttötapaukset

Uusien hahmokomponenttien lisääminen:

Sivustolta löytyy adminin käytettäväksi jokaiselle komponenttityypille erikoistunut lomake, jonka avulla uudet tiedot voidaan lisätä. Jokaiseen komponenttiin liittyy aina tieto mistä säännöstöstä (Rule Set) se on peräisin. Tästä syystä säännöstöt on aina lisättävä ennen niihin kuuluvien komponenttien lisäämistä.

Hahmokomponenttien muokkaaminen:

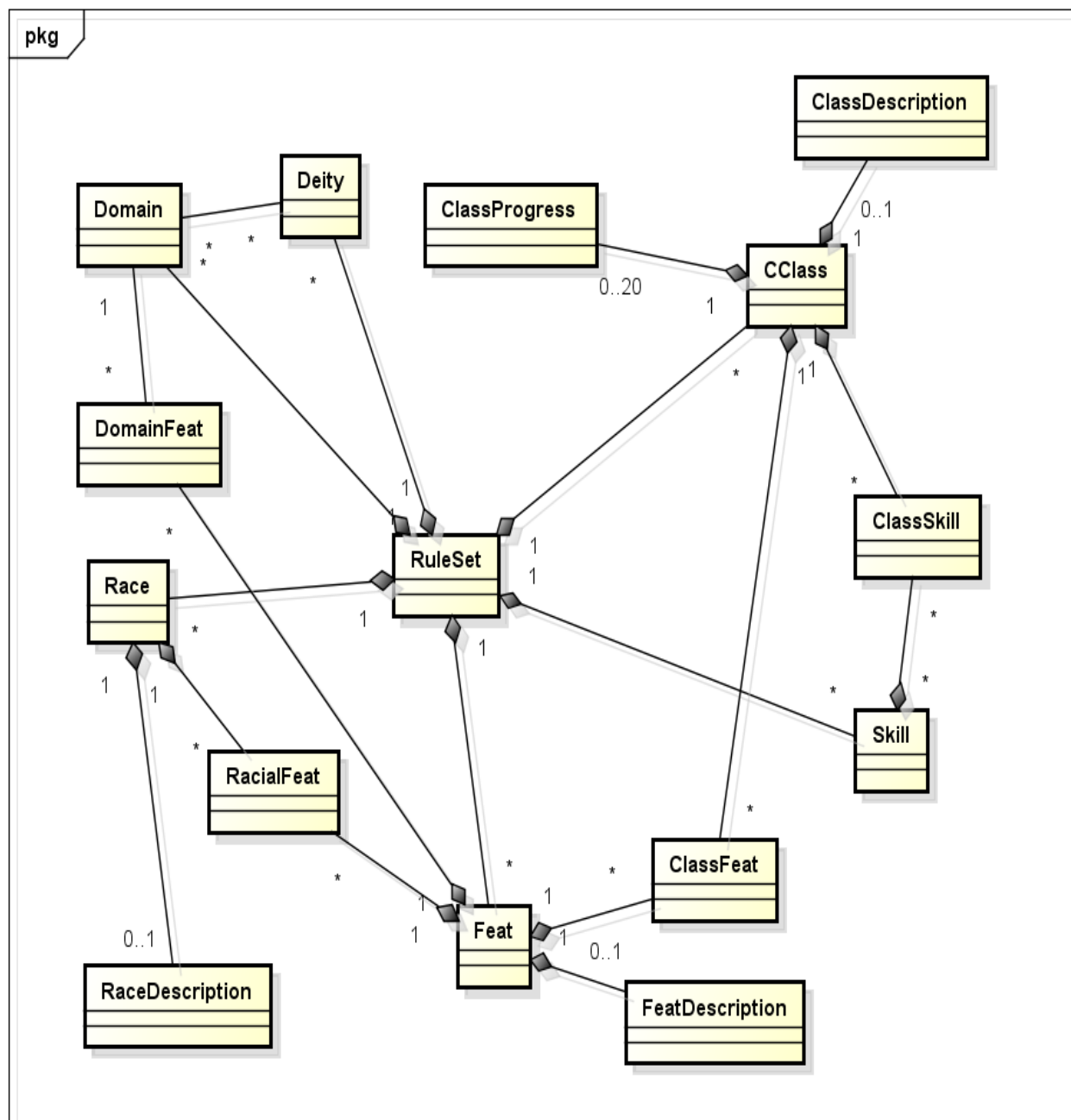
Adminin on mahdollista esim. lisätä vaatimuksia olemassa oleviin hahmokomponentteihin, sekä tarpeen tullen korjata virheellisiä tietoja.

Uusien säännöstöjen luominen:

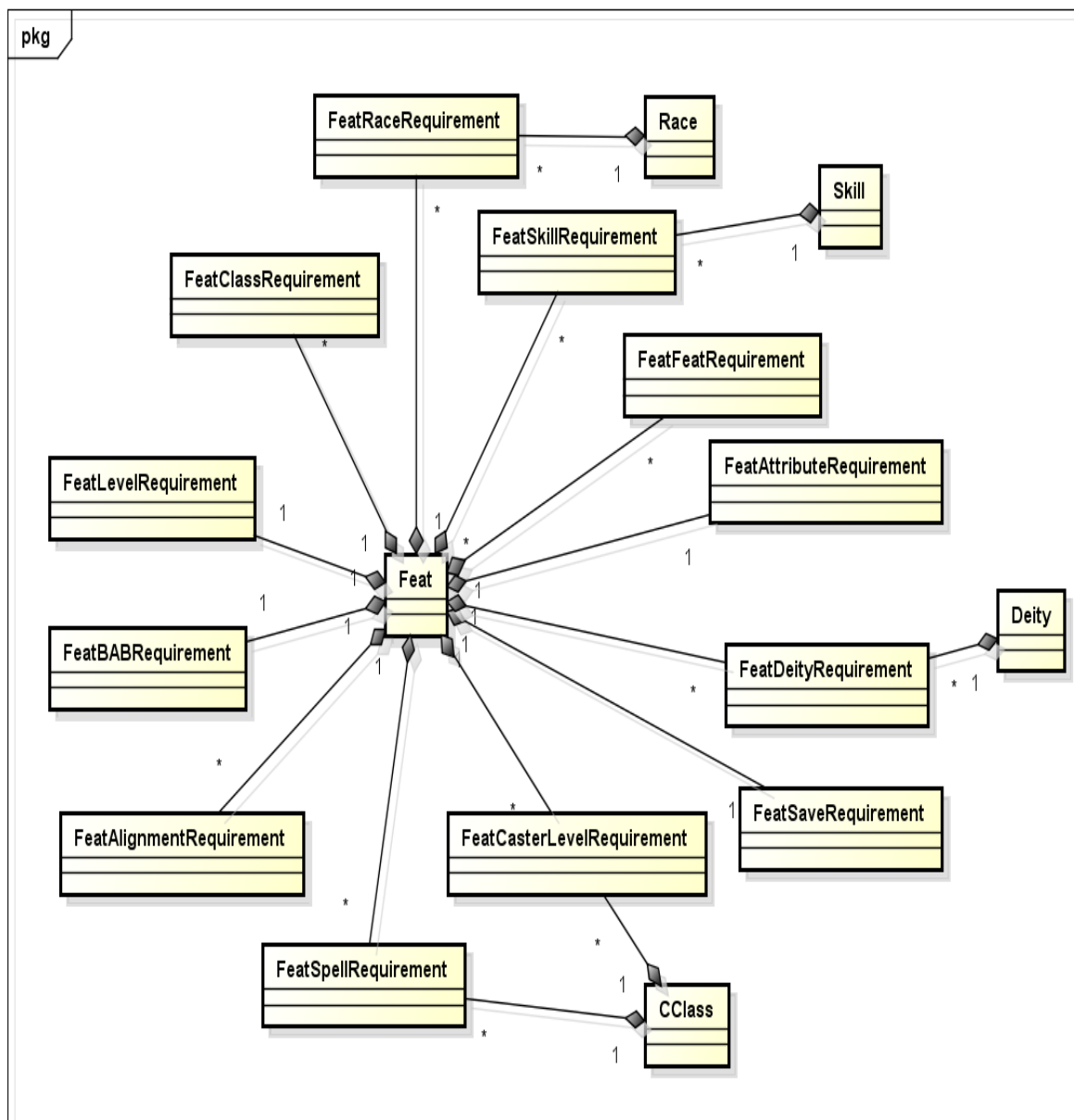
Admin voi lisätä tietokantaan uusia virallisia säännöstöjä ja halutessaan myös omiaan.

## Järjestelmän tietosisältö

Tietosisältö jaettuna useampaan kaavioon laajuuden vuoksi. Ensimmäisenä hahmokomponenttien välisiä yhteyksiä.



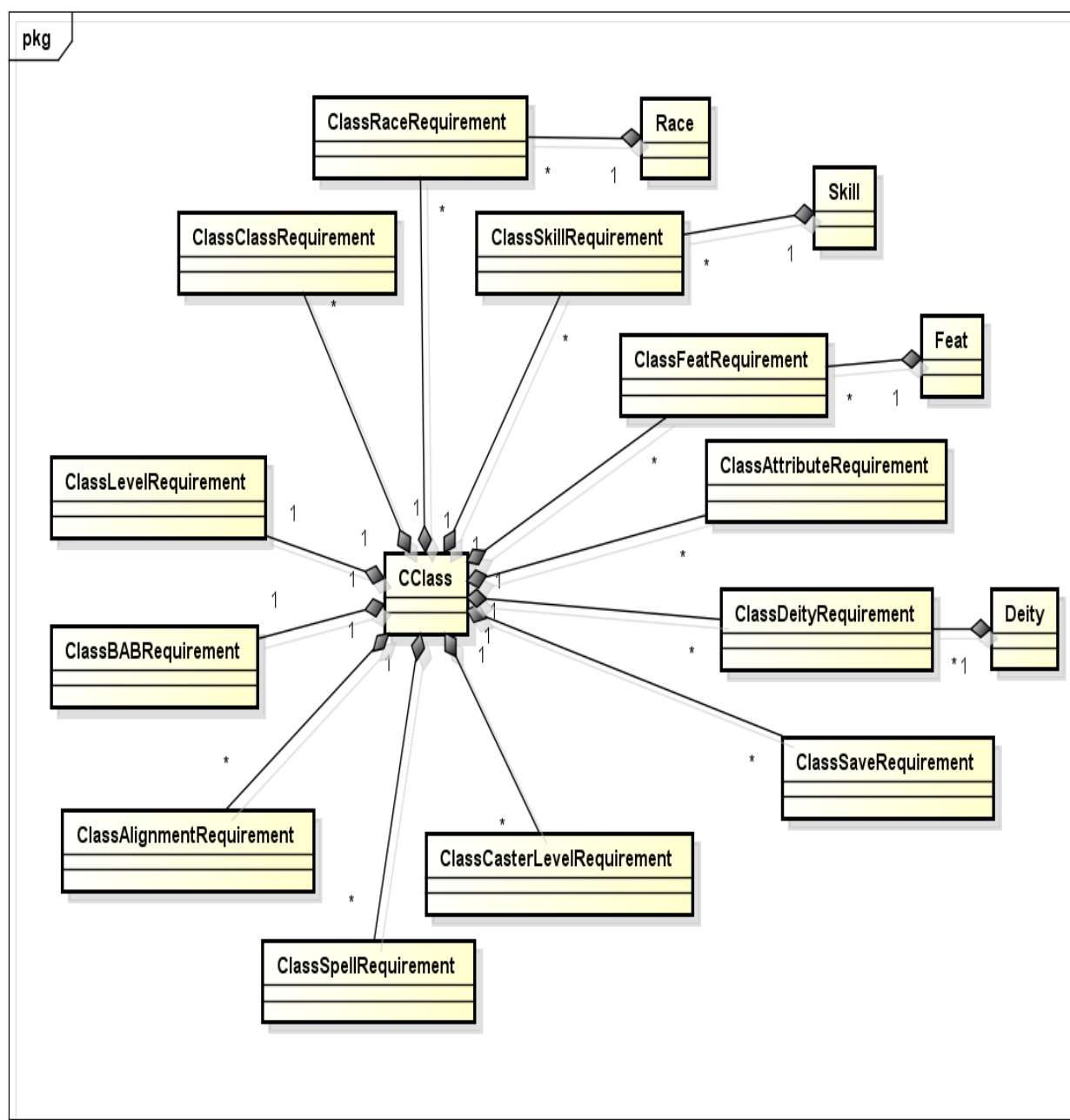
Tässä kaaviossa Featteihin liittyvät “vaatimukset”.



HUOM. Uusimmassa versiossa FeatAttributeRequiremetteja ja FeatSaveRequirementteja voi liittyä mielivaltainen määrä yhteen Feattiin.



Viimeisenä hahmoluokkiin liittyvät “vaatimukset”.



Kuten kaaviosta on nähtävissä, Feateillä ja CClasseilla on lähes identtiset vaatimustyytit, ja vaatimuksia on suhteellisen valtaisa määrä.

## Tietokohteiden tietosisältöjen kuvaukset

Huom. RuleSettiä lukuunottamatta kaikissa tauluissa on tällä hetkellä käytössä automaattisesti generoitu avain. Kaikki taulut generoidaan javan Entityjen pohjalta, joten samoin arvojen maksimirajat ovat javan Stringin, Integerin ja Longin määrittämiä.

### Tietokohde: RuleSet

Attribuutti	Arvojoukko	Kuvailu
name	varchar(255)	Lyhenne, RuleSetin avain
fullname	varchar(255)	Nimi kokonaisuudessaan

RuleSet on olennainen pohja, josta käytännössä kaikki muut taulut ovat riippuvaisia.

### Tietokohde: Race

id	bigint	Automaattisesti generoitu avainarvo
name	varchar(255)	Rodun nimi
ruleSet	varchar(255)	Säännöstö mihin kuuluu
STR	int	Strength-bonus
DEX	int	Dexterity-bonus
CON	int	Constitution-bonus
INTG	int	Intelligence-bonus
WIS	int	Wisdom-bonus
CHA	int	Charisma-bonus

Race liittyy johonkin RuleSettiin, ja jokaiseen rotuun voi liittää halutun määrän RacialFeatteja.

### Tietokohde: RacialFeat

race	bigint	Race mihin RacialFeat liittyy
feat	bigint	Feat mihin RacialFeat liittyy

RacialFeatin tehtävä on käytännössä liittää haluttu määrä Featteja johonkin Raceen.

**Tietokohde: CClass**

Attribuutti	Arvojoukko	Kuvailu
name	varchar(255)	CClassin nimi
ruleSet	varchar(255)	Säännöstö mihin kuuluu
*Required	boolean	Onko vaatimus olemassa

CClasseihin ja Featteihin liittyy molempiin suuri määrä erilaisia -Requirement nimettyjä tauluja jotka määrittelevät näiden vaatimuksia. CClass ja Feat on tällä hetkellä asetettu niin, että niistä näkee onko niille asetettu Requirementteja.

**Tietokohde: Feat**

Featin tietosisältö on sama kuin CClassin.

**Tietokohde: ClassFeat**

cclass	bigint	CClass mihin viitataan
feat	bigint	Feat mihin viitataan
lvl	int	Vaadittu taso

ClassFeat yhdistää RacialFeatin tavoin Featteja CClasseihin.

**Tietokohde: Skill**

name	varchar(255)	Skillin nimi
ruleSet	varchar(255)	RuleSet mihin kuuluu

Skillit liitetään ClassSkillin kautta CClasseihin.

**Tietokohde: ClassSkill**

Attribuutti	Arvojoukko	Kuvailu
cclass	bigint	CClass mihin viitataan
skill	bigint	Skill mihin viitataan
type	int	Skillin tyyppi, vaikuttaa hahmonrakentamisessa

ClassSkill toimii RacialFeatin ja ClassFeatin tavoin. Jokainen ClassSkill yhdistää yhden CClassin yhteen Skilliin, ja määrittää tälle yhteydelle tyyppin.

**Tietokohde: Deity**

name	varchar(255)	Deityn nimi
alignment	int	Deityn alignment
ruleSet	varchar(255)	RuleSet mihin kuuluu

Hahmonluonnissa hahmolle voi valita Deityn, ja mahdollisuudet määräytyvät hahmon alignment-arvon ja Deityjen alignment-arvojen perusteella.

**Tietokohde: DDomain**

name	varchar(255)	Domainin nimi
ruleSet	varchar(255)	RuleSet mihin kuuluu

DDomain voi liittyä joihinkin Deityihin ja Featteihin DomainFeatin kautta.

**Tietokohde: DomainFeat**

domain	bigint	DDomain mihin viitataan
feat	bigint	Feat mihin viitataan

Tietyillä CClasseilla on ominaisuutena se, että ne saavat Deitystä riippuen tiettyjä Featteja, jolloin DomainFeateilla voidaan halutusti määrittää kuhunkin Deityyn liittyvät Featit.

**Tietokohde: ClassProgress**

Attribuutti	Arvojoukko	Kuvailu
cclass	bigint	CClass mihin viitataan
BAB	int	Base attack bonus -arvo
hitDice	int	Hahmon hp-saannin määritin
casterLevel	int	Hahmon CClass-kohtainen caster level -määritin
spellLevel	int	Hahmon CClass-kohtainen max spell level -määritin
fortSave	int	Hahmon CClass-kohtainen fortitude saving throw
willSave	int	Hahmon CClass-kohtainen will saving throw
reflSave	int	Hahmon CClass-kohtainen reflex saving throw
skillPoints	int	Hahmon CClass-kohtainen skillpoint -saanto
lvl	int	Hahmon leveli CClassissa

Hahmonluonnin kannalta kenties olennaisin taulu, minkä kautta hahmon tasoa nostaessa määritetään hahmon uusia parametrejä, joita taas käytetään CClassien ja Feattien Requirementtien ratkaisemiseen.

**Tietokohde: \*Requirement**

Requirementtejä on kahta päätyyppiä, Class\*Requirement ja Feat\*Requirement, (missä \* on jokin vaadittu arvotyyppi). Requirementteissa on aina attribuuttina päätyypin olion avain (bigint ja \* tyylistä riippuen CClass/Feat/Skill/Race/Deity avain (bigint), AttributeRequirementissa on int-arvoista attributea, SaveRequirementissa 3, Level ja BABRequirementteissa 1. Poikkeuksena on ClassLevel ja SpellRequirementit, joissa on attribuutteina CClassin avain(bigint) sekä yksi int-arvo. Nämä taulut ovat olennaisia tulevassa hahmonluomisominaisuudessa.

**Tietokohde: ClassDescription, FeatDescription, RaceDescription**

cclass/feat/race	bigint	Tyypikohtainen avain
description	varchar(255)	Komponentin kuvaus

## Järjestelmän yleisrakenne

Servletit on jaettu karkeasti avustaviin servletteihin (Servlets.utility), joihin kuuluu esim. Loginin hoitaminen sekä erilaisten lisäyssivujen oliolistojen tarjoaminen. Varsinaiset tietokantaan lisäämistapahtumat käsitellään Servlets.additionServlets-hakemiston servleteissä, poistaminen vastaavasti .removalServlets-hakemistossa. Servlets.managers sisältää tietokannan varsinaisten hallintosivujen etenemistoiminnot. Vastaavasti .search on tarkoitettu hakuservleteille. Erilaisen apuluokat löytyvät ComponentManager-hakemistosta

Tiedostonimet ovat pääasiassa aina niin, että jokainen varsinainen erillinen sana alkaa isolla kirjaimella.

## Järjestelmän komponentit

Kaikki lisäys- ja poisto- sekä hakujsp-sivut käyttävät <http://java.sun.com/jsp/jstl/core> c-kirjastoa.

### JSP-sivut:

index.jsp

Järjestelmän etusivu, sisältää yksinkertaisen hakutoiminnon, joka käyttää SearchByNameServletiä ja ShowInfoServletiä.

publicnav.jsp

Julkisen puolen navigaatiopalkin koodi, joka tarjoaa mahdollisuuden kirjautumiseen ja kirjautuneelle käyttäjälle mahdollisuuden edetä hallinnon pääsivulle.

Login.jsp

Kirjautumissivu, järjestelmässä on käyttötarkoituksen vuoksi vain yksi käyttäjä, eikä tällä hetkellä lisäämismahdollisuuksia (joskin toteutus on tarvittessa tehtävissä).

Logout.jsp

Kirjaa käyttäjän ulos ja ohjaa takaisin index.jsp:hen.

LoginCheck.jsp

Kaikilla hallinnon sivuilla käytettävä koodinpätkä, joka tarkistaa onko käyttäjä kirjautunut ja jos ei niin ohjaa Login.jsp:hen.

adminnav.jsp

Hallinnon puolen sivujen navigaatiopalkki, mikä sisältää uloskirjautumisen lisäksi pääsyn kaikkiin hallinnon toimintoihin.

AddNewContent.jsp

Hallinnon pääsivu, missä on lyhykäiset ohjeet siitä, missä mitkäkin toiminnallisuudet ovat.

AddNewRuleSet.jsp

Mahdollistaa RuleSettien lisäämisen tietokantaan. Saa AddNewRuleSetServletiltä listan olemassaolevista RuleSeteistä, joka on nähtävillä, ja lähettää lisäämislomakkeensa eteenpäin AddRuleSetServletille.

AddNewClass.jsp

Mahdollistaa CClassien lisäämisen tietokantaan. Saa AddNewClassServletiltä listan olemassaolevista CClasseista, joka on nähtävillä, ja lähettää lisäämislomakkeensa eteenpäin AddClassServletille.

#### AddNewRace.jsp

Mahdollistaa Racejen lisäämisen tietokantaan. Saa AddNewRaceServletiltä listan olemassaolevista Raceista, joka on nähtävillä, ja lähettää lisäämislomakkeensa eteenpäin AddRaceServletille.

#### AddNewFeat.jsp

Mahdollistaa Feattien lisäämisen tietokantaan. Saa AddNewFeatServletiltä listan olemassaolevista Feateista, joka on nähtävillä, ja lähettää lisäämislomakkeensa eteenpäin AddFeatServletille.

#### AddNewSkill.jsp

Mahdollistaa Skillien lisäämisen tietokantaan. Saa AddNewSkillServletiltä listan olemassaolevista Skilleistä, joka on nähtävillä, ja lähettää lisäämislomakkeensa eteenpäin AddSkillServletille.

#### AddNewDeity.jsp

Mahdollistaa Deityjen lisäämisen tietokantaan. Saa AddNewDeityServletiltä listan olemassaolevista Deityistä, joka on nähtävillä, ja lähettää lisäämislomakkeensa eteenpäin AddDeityServletille.

#### AddNewDomain.jsp

Mahdollistaa DDomainien lisäämisen tietokantaan. Saa AddNewDomainServletiltä listan olemassaolevista DDomaineista, joka on nähtävillä, ja lähettää lisäämislomakkeensa eteenpäin AddDomainServletille.

#### Manage.jsp

Erilaisille yhteyksien määrittelysivuille ja kaikille poistosivuille polut tarjoava sivu.

#### ManageClassProperties.jsp

Mahdollistaa kaikkien CClassiin liittyvien olioiden määrittelyn ja poistamisen, sekä CClassin poistamisen tietokannasta. Sivulla on mahdollista valita CClass ja sitten toiminto jonka sen kohdalla haluaa toteuttaa. Saa ManageClassesServletiltä listan kaikista CClasseista. Lähettää lomakkeensa ManageClassServletille, joka sitten redirectaa oikealle sivulle riippuen painetusta submit-näppäimestä.

#### AddNewClassSkill.jsp

Mahdollistaa CClasseja ja Skillejä yhdistävien ClassSkill-olioiden lisäämisen tietokantaan, myös poistaminen tapahtuu täällä. AddNewClassSkillServlet tarjoaa listan kaikista kyseisen CClassin ClassSkilleistä ja Skilleistä joihin ei ole määritetty yhteyttä. Lomake lähetetään AddClassSkillServletille joka hoitaa tietokantaan lisäämisen ja ohjaa sitten takaisin tänne. Poistaessa lomake lähettää tiedot RemoveClassSkillServletille, joka ohjaa taas takaisin tänne.



#### AddNewClassFeat.jsp

Mahdollistaa CClasseja ja Featteja yhdistävien ClassFeat-olioiden lisäämisen tietokantaan, myös poistaminen tapahtuu täällä. AddNewClassFeatServlet tarjoaa listan kaikista kyseisen CClassin ClassFeateista ja Feateista joihin ei ole määritetty yhteyttä. Lomake lähetetään AddClassFeatServletille joka hoitaa tietokantaan lisäämisen ja ohjaa sitten takaisin tänne. Poistaessa lomake lähettää tiedot RemoveClassFeatServletille, joka ohjaa taas takaisin tänne.

#### AddClassRequirements.jsp

Täällä toteutetaan kaikkien Class\*Requirementtien lisääminen ja poisto. AddClassRequirementsServlet tarjoaa listat kaikista asetetuista Requirementteista ja näyttää täytettävät kentät vain jos CClassille on tarkoitus luoda niiden tyyppisiä Requirementteja. Sama servlet hoitaa syötteiden tarkistamisen ja lisäämisen tietokantaan. Poistaessa lomakkeet lähetetään RemoveClassRequirementServletille joka tarkistaa poistettavan Requirementin tyyppin ja hoitaa sitten tietokantatapahtumat.

#### SetClassProgress.jsp

Tämä sivu mahdollistaa ClassProgress-olioiden automaattisen generoimisen annettujen parametrien perusteella käyttämällä SetClassProgressServletiä, joka tarkistaa syötteet ja sitten asettaa oikeat arvot luotaville ClassProgress-olioille. Poistaminen tapahtuu lähettämällä lomake RemoveClassProgressServletille, joka poistaa kaikki CClassiin liittyvät ClassProgressit.

#### AddNewClassDescription.jsp

Mahdollistaa ClassDescriptionin lisäämisen CClass-kohtaisesti. AddDescriptionServlet saa tiedon siitä, mille oliotyypille ollaan Descriptionia luomassa, ja näyttää jo-olemassa olevan Descriptionin luo uuden lomakkeen tietojen pohjalta. Poisto tapahtuu lähettämällä lomake RemoveClassDescriptionServletille.

#### RemoveClass.jsp

Mahdollistaa CClassin ja kaikkien siihen viittaavien tietokannan rivien poistamisen. Tiedot lähetetään ConfirmRemovalServletille, joka käyttää apunaan ClassRemover-oliota, joka käy läpi kaikki kyseisen CClassin yhteydet ja poistaa löydökset ja viimeisenä CClassin itsensä.

### ManageFeatProperties.jsp

Mahdollistaa kaikkien Featteihin liittyvien olioiden määrittelyn ja poistamisen, sekä Feattien poistamisen tietokannasta. Sivulla on mahdollista valita Feat ja sitten toiminto jonka sen kohdalla haluaa toteuttaa. Saa ManageFeatsServletiltä listan kaikista Feateista. Lähettää lomakkeensa ManageFeatServletille, joka sitten redirectaa oikealle sivulle riippuen painetusta submit-näppäimestä.

### AddFeatRequirements.jsp

Täällä toteutetaan kaikkien Feat\*Requirementtien lisääminen ja poisto.

AddFeatRequirementsServlet tarjoaa listat kaikista asetetuista Requirementteista ja näyttää täytettävät kentät vain jos Featille on tarkoitus luoda niiden tyyppisiä Requirementteja. Sama servlet hoitaa syötteiden tarkistamisen ja lisäämisen tietokantaan. Poistaessa lomakkeet lähetetään RemoveFeatRequirementServletille joka tarkistaa poistettavan Requirementin tyyppin ja hoitaa sitten tietokantatapahtumat.

### AddNewFeatDescription.jsp

Mahdollistaa FeatDescriptionin lisäämisen Feat-kohtaisesti. AddDescriptionServlet saa tiedon siitä, mille oliotyyppille ollaan Descriptionia luomassa, ja näyttää jo-olemassa olevan Descriptionin luo uuden lomakkeen tietojen pohjalta. Poisto tapahtuu lähettämällä lomake RemoveFeatDescriptionServletille.

### RemoveFeat.jsp

Mahdollistaa Featin ja kaikkien siihen viittaavien tietokannan rivien poistamisen. Tiedot lähetetään ConfirmRemovalServletille, joka FeatRemover-oliota käyttäen käy läpi kaikki kyseisen Featin yhteydet ja poistaa löydökset ja viimeisenä Featin itsensä.

### ManageRaceProperties.jsp

Mahdollistaa kaikkien Raceihin liittyvien olioiden määrittelyn ja poistamisen, sekä Racejen poistamisen tietokannasta. Sivulla on mahdollista valita Race ja sitten toiminto jonka sen kohdalla haluaa toteuttaa. Saa ManageRacesServletiltä listan kaikista Raceista. Lähettää lomakkeensa ManageRaceServletille, joka sitten redirectaa oikealle sivulle riippuen painetusta submit-näppäimestä.

### AddNewRacialFeat.jsp

Mahdollistaa RacialFeattien lisäämisen Raceille. AddNewRacialFeatServlet tarjoaa listat kaikista Racelle asettamattomista ja asetetuista Feateista. Lomake lähetetään AddRacialFeatServletille, joka hoitaa lisäämisen. Poistolomake vastaavasti lähettää valitun RacialFeatin tiedot RemoveRacialFeatServletille, joka hoitaa poiston ja redirectaa takaisin tänne.

#### AddNewRaceDescription.jsp

Mahdollistaa RaceDescriptionin lisäämisen Race-kohtaisesti.  
AddDescriptionServlet saa tiedon siitä, mille oliotyypille ollaan Descriptionia luomassa, ja näyttää jo-olemassa olevan Descriptionin luo uuden lomakkeen tietojen pohjalta. Poisto tapahtuu lähettämällä lomake RemoveRaceDescriptionServletille.

#### RemoveRace.jsp

Mahdollistaa Racen ja kaikkien siihen viittaavien tietokannan rivien poistamisen. Tiedot lähetetään ConfirmRemovalServletille, joka käyttää apunaan RaceRemover-oliota, joka käy läpi kaikki kyseisen Racen yhteydet ja poistaa löydökset ja viimeisenä Racen itsensä.

#### ManageDeityProperties.jsp

Mahdollistaa kaikkien Deityihin liittyvien olioiden määrittelyn ja poistamisen, sekä Deityjen poistamisen tietokannasta. Sivulla on mahdollista valita Deity ja sitten toiminto jonka sen kohdalla haluaa toteuttaa. Saa ManageDeitiesServletiltä listan kaikista Deityistä. Lähettää lomakkeensa ManageDeityServletille, joka sitten redirectaa oikealle sivulle riippuen painetusta submit-näppäimestä.

#### AddNewDeityDomain.jsp

Mahdollistaa DeityDomainien lisäämisen Deitylle. AddNewDeityDomainServlet tarjoaa listat kaikista Deitylle asettamattomista ja asetetuista DDomaineista. Lomake lähetetään AddDeityDomainServletille, joka hoitaa lisäämisen. Poistolomake vastaavasti lähettää valitun DeityDomainin tiedot RemoveDeityDomainServletille, joka hoitaa poiston ja redirectaa takaisin tänne.

#### RemoveDeity.jsp

Mahdollistaa Deityn ja kaikkien siihen viittaavien tietokannan rivien poistamisen. Tiedot lähetetään ConfirmRemovalServletille, joka käyttää apunaan DeityRemover-oliota, joka käy läpi kaikki kyseisen Deityn yhteydet ja poistaa löydökset ja viimeisenä Deityn itsensä.

#### ManageDomainProperties.jsp

Mahdollistaa kaikkien DDomaineihin liittyvien olioiden määrittelyn ja poistamisen, sekä DDomainien poistamisen tietokannasta. Sivulla on mahdollista valita DDomain ja sitten toiminto jonka sen kohdalla haluaa toteuttaa. Saa ManageDomainsServletiltä listan kaikista DDomaineista. Lähettää lomakkeensa ManageDomainServletille, joka sitten redirectaa oikealle sivulle riippuen painetusta submit-näppäimestä.

#### AddNewDomainFeat.jsp

Mahdollistaa DomainFeattien lisäämisen DDomainille. AddNewDomainFeatServlet tarjoaa listat kaikista DDomainille asettamattomista ja asetetuista Feateista. Lomake lähetetään AddDomainFeatServletille, joka hoitaa lisäämisen. Poistolomake vastaavasti lähettää valitun DomainFeatin tiedot RemoveDomainFeatServletille, joka hoitaa poiston ja redirectaa takaisin tänne.

#### RemoveDomain.jsp

Mahdollistaa DDomainin ja kaikkien siihen viittaavien tietokannan rivien poistamisen. Tiedot lähetetään ConfirmRemovalServletille, joka käyttää apunaan DomainRemover-oliota, joka käy läpi kaikki kyseisen DDomainin yhteydet ja poistaa löydökset ja viimeisenä DDomainin itsensä.

#### ManageSkillProperties.jsp

Mahdollistaa Skillien hallinnoinnin, johon tällä hetkellä kuuluu vain poistaminen. Saa ManageSkillsServletiltä listan kaikista Skilleistä ja lähettää lomakkeensa ManageSkillServletille, joka ohjaa oikealle sivulle submit-painikekohtaisesti.

#### RemoveSkill.jsp

Mahdollistaa Skillin ja kaikkien siihen viittaavien tietokannan rivien poistamisen. Tiedot lähetetään ConfirmRemovalServletille, joka käyttää apunaan SkillRemover-oliota, joka käy läpi kaikki kyseisen Skillin yhteydet ja poistaa löydökset ja viimeisenä Skillin itsensä.

#### ManageRuleSetProperties.jsp

Rule Settien hallinnointisivu, jonka kautta on mahdollista poistaa suurimpia yhteyskokonaisuuksia. Saa ManageRuleSetsServletiltä tiedon kaikista olemassaolevista RuleSeteistä ja lomake lähetetään ManageRuleSetServletille, joka ohjaa eteenpäin oikeaan paikkaan.

#### RemoveRuleSet.jsp

Mahdollistaa RuleSetin ja kaikkien siihen viittaavien tietokannan rivien poistamisen. Lomake lähetetään ConfirmRuleSetRemovalServletille, joka ohjaa vielä eteenpäin.

#### RuleSetDeletionConfirmation.jsp

Koska RuleSettien poistamisella voi olla kauaskantoisia seurauksia, poistamiselle on viimeinen varmistus, joka vaatii erillisen salasanan syöttämisen ja hyväksymisen. Lomake menee eteenpäin ConfirmRemovalServletille, joka käyttää RuleSetRemover-oliota kaikkien RuleSetin yhteyksien poistamiseen tietokannasta.

## CSS:

Miltei kaikki jsp-sivut käyttävät MainCSS-nimistä css-tiedostoa.

## Servletit:

.utility

AddNew -alkuiset servletit toimivat aina nimeänsä vastaavien jsp-sivujen apukomponentteina, siis tarjoavat listat olemassaolevista asiaankuuluvista tietokannan riveistä.

LoginServlet käsittelee kirjautumisyrityksen, ja asettaa käyttäjän kirjautuneeksi jos tiedot ovat kohdallaan.

.addition

Kaikki tämän hakemiston servletit ottavat vastaan lomakkeita ja tarkistavat niiden sisältämien parametrien oikeellisuuden. Jos sopivat tiedot on annettu, eikä lisättäväksi esitettävää riviä löydy tietokannasta valmiiksi, nämä Servletit suorittavat määrätyn lisäämistapahtuman. Joillakin servleteilla, kuten SetClassProgressServletillä, on paljon metodeja oikeiden lisättävien arvojen generoimiseksi.

.managers

Monikkonimiset servletit tarjoavat omille Manage\*Properties.jsp -sivuilleen listat olemassaolevista kyseisen sivun hallinnoimista oliotyypeistä. Yksikkönimiset ottavat vastaan jsp-sivujen lomakkeet ja ohjaavat painetusta submit-näppäimestä riippuville sivuille.

.removal

Sisältää kaikki poistotapahtumia käsittelevät servletit, RemoveClass/Feat/Skill/Race/Deity/Domain/RuleSetServletit toimivat vastaavien Remove\*.jsp -sivujen apureina, joista poistotapahtumat etenevät ConfirmRemovalServletille. Muiden oliotyyppien Remove\*Servletit käsittelevät yksittäisiä poistotapahtumia, eivätkä tarvitse siihen ConfirmRemovalServlettiä.

.search

Erilaiset hakupalveluservletit, tällä hetkellä vain etusivun pikahaun suoritus ja lisätiedot.

## Apuluokat:

### ComponentManager

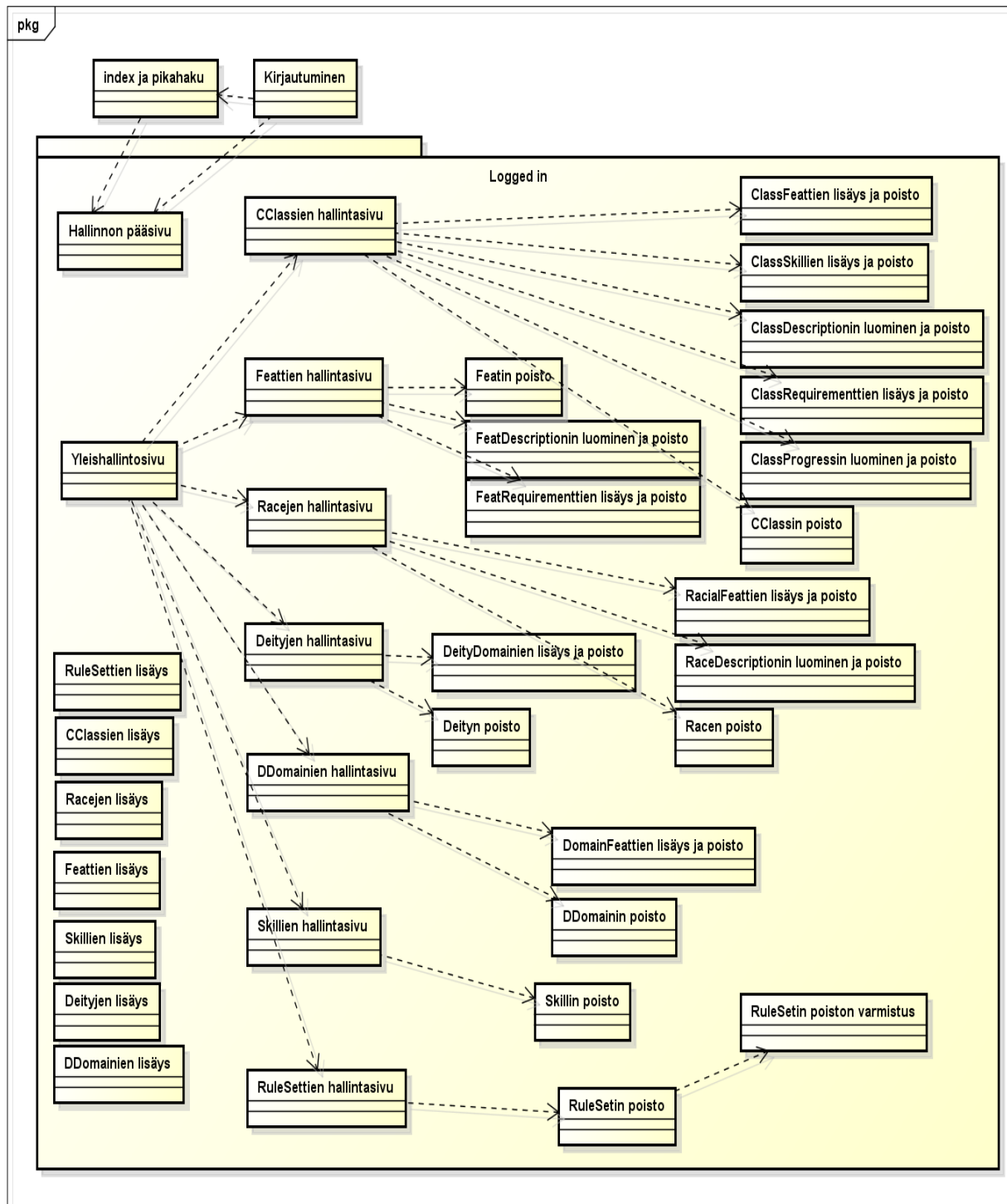
Tämä hakemisto sisältää kaikki tietokannan lisäys-, poisto- ja hakutoimintoja suorittavat luokat. Kaikki Remover-päätteiset luokat ovat vastuussa suurempien kokonaisuuksien poistoista. EM luokka käyttää EntityManagerFactoryä ja EntityManageria erilaisten queryjen suorittamiseksi. EM suorittaa kaikki yksittäiset lisäys- ja poistotapahtumat, sekä hakutoimintojensa vuoksi se on käytössä käytännössä jokaisessa servletissä.

## Entityluokat:

### Components

Tämä hakemisto ja sen alihakemistot sisältävät kaikki tietokantaan lisättävää tietoa määrittelevät luokat.

## Käyttöliittymä



Logged in -puolella navigaatiopalkin kautta pääsee suoraan kaikille kaavion vasemman laidan sivuille, sekä uloskirjautumiseen. Julkisen puolen navigaatiopalkista pääsee sisäänkirjautumiseen, ja kirjautuneen uloskirjautumiseen ja hallinnon pääsivulle.

## Asennustiedot

Sovellus on asennettu tällä hetkellä Helsingin yliopiston tietojenkäsittelytieteen laitoksen palvelimelle, jossa se toimii Tomcatilla. Jos sovelluksen haluaa saada toimimaan halutussa ympäristössä, täytyy ensin asettaa sille NetBeansin kautta oikeat tietokanta-ajurit jos käytössä ei ole PostGreSQL. Tämän lisäksi täytyy muokata sovelluksen persistence.xml -tiedostoon halutun tietokannan url, käyttäjätunnus, ajuri sekä salasana tiedot. Kun nämä muutokset on suoritettu, täytyy sovellus enää Clean&Buildata, jonka jälkeen sen saa toimimaan yksinkertaisesti siirtämällä buidatun dist-kansion DnDCharBuilder.war tiedoston palvelimen tomcatin webapps-kansioon.



## Käynnistys- / Käyttöohje

Sovellus on käytettävissä osoitteessa [t-hxvehmas.users.cs.helsinki.fi/DnDCharBuilder/](http://t-hxvehmas.users.cs.helsinki.fi/DnDCharBuilder/)  
Kirjautumiseen tarvittavat tunnukset ovat:

Username: DM

Password: DM

Sivun käyttäjän olisi oletuksena tarkoitus tietää, mitä mikäkin tekee, mutta koska tämä ei tässä tapauksessa ole välttämättä mahdollista, niin sivulla on jonkin verran ohjeistusta ja tässä vielä yleinen toimintajärjestysohje.

RuleSetit on lisättävä ennen mitään muuta, sillä kaikki on niistä riippuvaista. Sääntöjä on turha yrittää lisätä jos niillä ei ole mitään lähdettä. Lisäyskentissä ehkä hämäävä Source siis tarkoittaa RuleSetin lyhennettä. RuleSetin lisäämisen jälkeen voi vapaasti luoda siihen liittyviä muita pääkomponentteja, mihin on suora pääsy hallinnon navigaatiopalkilta. Kun pääkomponentteja on luotu, voidaan niiden yhteyksiä määritellä sekä poistoja suorittaa navigaatiopalkin kautta Manage Componentsista.

Otan myös mielelläni vastaan palautetta siitä, jos jotkin sivut tai toiminnot hajoavat jostain syystä. Olen testannut niin paljon kuin olen ehtinyt ja erilaisia tarkituksia on paljon, mutta toiminnallisuuksien määrän vuoksi virhemahdollisuuksia voi aina löytyä lisää.