

hyväksymispäivä arvosana

arvostelija

Suunnitteludokumentti
Vaatimusanalyysin hallintatyökalu

Ranta Sinikka, Helminen Marja, Marttila Jussi, Muhonen Jussi,
Virkkala Timo, Vuorinen Matti, Laitinen Jarmo

Helsinki 13.4.2005

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Tiedekunta/Osasto — Fakultet/Sektion — Faculty		Laitos — Institution — Department	
Matemaattis-luonnontieteellinen tdk.		Tietojenkäsittelytieteen laitos	
Tekijä — Författare — Author			
S. Ranta, M. Helminen, J. Marttila, J. Muhonen, T. Virkkala, M. Vuorinen, J. Laitinen			
Työn nimi — Arbetets titel — Title			
Vaatimusanalyysin hallintatyökalu			
Oppiaine — Läroämne — Subject			
Tietojenkäsittelytiede			
Työn laji — Arbetets art — Level		Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Suunnitteludokumentti		13.4.2005	72 sivua + 108 liitesivua
Tiivistelmä — Referat — Abstract			
<p>Vaativa on järjestelmä vaatimusanalyysin hallintaan. Sen avulla voidaan seurata ja hallita projektien vaatimuksia ja käyttötapauksia. Se on etupäässä suunniteltu Tietojenkäsittelytieteen laitoksen ohjelmistotuotantoprojektien käyttöön.</p> <p>Vaativa-järjestelmällä voidaan kerätä, kartoittaa, hallita ja seurata järjestelmän vaatimuksia ja käyttötapauksia, sekä niiden yhteyksiä. Lisäksi järjestelmän avulla voidaan tuottaa vaatimudokumentin runko. Vaativa-järjestelmä on laajennettavissa liitännäisillä, joilla voidaan lisätä tapoja syöttää ja tulostaa tietoa. Se käyttää tietovarastona JDBC-yhteensopivaa tietokantaa.</p> <p>Tässä dokumentissa esitellään Vaativa-järjestelmän toteutussuunnitelma. Esiteltäviä tietoja ovat järjestelmän arkkitehtuuri, tietokantarakenne, luokkarakenne, tiedon kulku järjestelmässä sekä käyttöliittymä. Lisäksi dokumentissa esitellään järjestelmän tuottaman vaatimudokumentin rakenne ja joitakin järjestelmän käyttöskenaarioita.</p> <p>ACM Computing Classification System (CCS): D.2.1 [Software Engineering]: Requirements/Specifications, D.2.2 [Software Engineering]: Design Tools and Techniques, J.6. [Computer Applications]: Computer-aided Engineering</p>			
Avainsanat — Nyckelord — Keywords			
Ohjelmistotuotantoprojekti, suunnitteludokumentti			
Säilytyspaikka — Förvaringsställe — Where deposited			
Muita tietoja — övriga uppgifter — Additional information			

Sisältö

1	Johdanto	1
1.1	Dokumentin tarkoitus	1
1.2	Dokumentin rakenne	1
1.3	Sanasto	2
2	Järjestelmän kokonaiskuvaus	6
2.1	Toteutus- ja toimintaympäristö	6
3	Arkkitehtuurin kuvaus	7
3.1	Sisäinen arkkitehtuuri	7
3.1.1	Tiedonsaantikerros eli Data Access Layer	7
3.1.2	Liitännäisjärjestelmä eli Plugin system	8
3.1.3	Ytimen arkkitehtuuri	8
3.2	Ulkoinen arkkitehtuuri	8
3.3	Rajapinnat	9
3.3.1	Malli	9
3.3.2	Näkymä	13
3.3.3	Tiedonsaantitaso	14
3.3.4	Liitännäisjärjestelmä	17
4	Tietokanta	17
4.1	Tietokantataulut	19
4.1.1	Attachment	19
4.1.2	Connection	20
4.1.3	Connection_type	21
4.1.4	Description	22
4.1.5	Entity	22
4.1.6	Entry_panel	23

4.1.7	Milestone	23
4.1.8	Priority	24
4.1.9	Project	24
4.1.10	Requirement	25
4.1.11	Requirement_to_Attachment	26
4.1.12	Requirement_to_Entity	27
4.1.13	Requirement_type	27
4.1.14	Stakeholder	27
4.1.15	Statistics	28
4.1.16	Status	28
4.1.17	Use_case	29
4.1.18	Use_case_to_Attachment	29
4.1.19	Use_case_to_Entity	30
4.1.20	Use_case_to_Stakeholder	30
4.1.21	User	30
4.1.22	User_per_Project	31
5	Tiedon kulku järjestelmässä	32
6	Komponenttien esittely (luokkien ja metodien kuvaus)	33
6.1	Ydin	33
6.1.1	Controller	33
6.1.2	DatabaseController	34
6.1.3	ViewController	34
6.1.4	View	34
6.1.5	DialogFactory	35
6.1.6	Model	35
6.2	Tiedonsaantikerros	35
6.2.1	DAOFactory	35

6.3	Liitännäisjärjestelmä	37
6.3.1	PluginManager	37
6.3.2	Plugin (rajapintaluokka)	37
6.3.3	InputPlugin (rajapintaluokka extends Plugin)	37
6.3.4	OutputPlugin (rajapintaluokka extends Plugin)	38
6.3.5	InputPanelPlugin (rajapintaluokka extends Plugin)	38
7	Käyttöliittymän kuvaus	38
7.1	Tekniikka	39
7.2	Käyttöliittymän ulkoasustandardit	39
7.3	Käyttöliittymän näkymien kuvaukset	41
7.3.1	Sovelluksen aloittaminen	41
7.3.2	Käyttäjän käyttöliittymä	46
7.3.3	Ylläpitäjän käyttöliittymä	69

Liitteet

1 Tuotettava vaatimusdokumentin runko

2 Skenaariot

3 Mallin dokumentaatio

1 Johdanto

Vaativa on työkalu vaatimusten hallintaan. Sen avulla voidaan kerätä, kartoittaa, hallita ja seurata järjestelmän vaatimuksia ja käyttötapauksia, sekä tuottaa runko vaatimusdokumentille. Lisäksi voidaan hallita vaatimusten yhteyksiä ja ristiriitoja, sekä seurata vaatimusten toteutumista projektin elinkaaren aikana.

Vaativa-järjestelmän tietosisältöä jäsennetään projekteina. Jokaisella projektilla voi olla useita käyttäjiä, jotka lisäävät ja muokkaavat projektin vaatimuksia ja käyttötapauksia. Käyttäjä voi myös olla jäsenenä useassa projektissa, sekä perustaa uusia projekteja. Järjestelmän tietokannasta huolehtivat järjestelmän ylläpitäjät.

Järjestelmään voidaan kirjata vaatimuksia ja käyttötapauksia vapaamuotoisena tekstinä, minkä lisäksi järjestelmään voidaan lisätä tuki strukturoiduille kuvauksille liitännäisten (*plug-in*) avulla. Myös eri muotoisten vaatimusdokumenttirunkojen lisääminen on mahdollista.

1.1 Dokumentin tarkoitus

Suunnitteludokumenttiin on koottu suunnitteluvaiheessa tehdyt ohjelman toteutusta koskevat päätökset. Dokumentti toimii projektiryhmän ohjeena teknisen toteutusvaiheen aikana. Suunnitteludokumentin tavoitteena on kuvata Vaativa-järjestelmä sellaisella tarkkuudella, että sen toteutus on suoraviivaista. Suunnittelun tuloksena saadaan kuvaukset toteutettavasta ohjelmistosta, arkkitehtuurista, tietokannoista, järjestelmän tarvitsemista ja tuottamista tiedoista, ohjelman eri osien ja ulkomaailman välisistä rajapinnoista, sekä järjestelmän komponenteista.

Suunnittelussa on otettu huomioon määrittelydokumentin luvussa 4 (Vaatimukset) luetelluista vaatimuksista prioriteetille 1 ja tärkeimmät prioriteetille 2 merkityt toiminnot. Prioriteetin 2 vaatimuksista toteutetaan se minimimäärä, joka tarvitaan prioriteetin 1 vaatimusten toteuttamiseen. Muista prioriteettien 2, 3 ja 4 toimintojen toteuttamisesta päätetään aikataulun salliessa toteutusvaiheen aikana.

1.2 Dokumentin rakenne

Suunnitteludokumentin ensimmäinen luku sisältää johdannon aiheeseen, kuvauksen dokumentin tarkoituksesta ja rakenteesta sekä dokumentin sanaston.

Luvussa kaksi esitellään järjestelmän kokonaiskuvaus toteutus- ja toimintaympäris-

töineen. Luvussa kolme esitellään järjestelmän sisäinen ja ulkoinen arkkitehtuuri, luvussa neljä tietokannan rakenne ja tietokantataulut ja luvussa viisi on kuvaus tiedon kulusta järjestelmässä.

Luvussa kuusi esitellään järjestelmän komponentit ja luokat. Luku seitsemän sisältää kuvauksen järjestelmän käyttöliittymästä ja näkymistä. Dokumentin liitteenä ovat kuvaus tuotettavasta suunnitteludokumentin rungosta sekä joitakin järjestelmän käyttöskenaarioita.

1.3 Sanasto

BLOB (Binary Large Object):

Relaatiotietokannoissa binääritiedon tallennukseen käytettävä tietotyyppi.

DAO (Data Access Object):

Olio, joka huolehtii tietokannassa olevan tiedon käsittelystä.

Dialogi:

Ohjelman yksinkertainen ikkuna, jonka kautta ohjelma kerää tietoa käyttäjältä tai antaa sitä.

Dokumenttirungon tuottajaliitännäinen:

Liitännäinen, joka määrittelee uuden dokumenttirungon.

Dokumenttirunko:

Tuotettavan suunnitteludokumentin malli.

Ensisijainen avain:

Tietokantataulun sarake tai sarakeyhdistelmä, jonka avulla taulun jokainen rivi voidaan yksilöidä.

Erikoistapaus:

Käyttötapausten normaalista suorituksesta poikkeava tilanne.

Etappi (Milestone):

Projektille tärkeä päivämäärä, johon mennessä tietyn vaatimus- ja käyttötapausten tulee olla määritellyssä tilassa.

Historiatiedot:

Projektin vaatimusten ja käyttötapausten kaikki aikaisemmat versiot.

Ikkuna:

Käyttöliittymän näkymä, joka avautuu erilliseksi laatikoksi ruudulle.

JDBC (Java Database Connectivity):

Javan tarjoama yleinen tietokantarajapinta.

Kohdeolio:

Järjestelmän osa, sidosryhmä tms. johon vaatimus tai käyttötapaus vaikuttaa.

Komponentti:

Ohjelman itsenäinen osa.

Käyttäjä:

Tarkoittaa peruskäyttäjää tai projektin omistajaa.

Käyttöliittymä:

Järjestelmän graafinen tai tekstipohjainen osa, jonka kautta ohjelmaa käytetään.

Käyttötapaus:

Kuvaus toiminnosta tai palvelusta, jonka järjestelmä toteuttaa, sekä tapahtuman onnistuneesta läpiviennistä ja mahdollisista variaatioista ja virhetilanteista.

Liitännäinen (Plug-in):

Järjestelmään jälkikäteen lisättävä osa, joka lisää järjestelmän toiminnallisuutta.

Liitännäisjärjestelmä:

Tarjoaa tavan laajentaa järjestelmän toiminnallisuutta myöhemmässä vaihessa, myös muiden kuin ohjelman kehittäjien toimesta.

MVC-malli (Model-View-Controller):

Käyttöliittymän suunnittelumalli. MVC-malli erottaa mallin (Model), näkymän (View) ja ohjaimen (Controller) toisistaan.

Näkymä:

Käyttöliittymän osa/ikkuna/sivu. Käyttöliittymässä voi olla monia eri näkymiä, jotka toteuttavat jonkin loogisen kokonaisuuden.

Ohjain:

Järjestelmän toimintaa ohjaava osajärjestelmä.

Olio (tietokantaosassa):

Olio on vaatimus tai käyttötapaus, kun sanaa käytetään tietokanta-osiossa.

Osajärjestelmä:

Ohjelman itsenäinen osa, jolla on tarkoin määritelty tehtävä.

Peruskäyttäjä:

Järjestelmään rekisteröitynyt projektin jäsen.

Projekti:

Kokonaisuus, jolla hallitaan vaatimuksia ja käyttötappauksia.

Projektin omistaja:

Peruskäyttäjä, joka on perustanut käsiteltävän projektin.

Projektin poistaminen:

Projektin kaikkien tietojen poistaminen järjestelmästä siten, että myös historiatiedot poistetaan.

Projektin sulkeminen:

Projektin asettaminen tilaan, jossa sen tiedot säilyvät, mutta niitä ei pääse muokkaamaan.

Rajapinta:

Kahden järjestelmän tai osajärjestelmän välinen liittymä, jonka kautta järjestelmät kommunikoivat keskenään.

Resoluutio:

Näytön tarkkuus.

Revisio:

Versionumero tiedolle, jonka historiatiedot säilytetään.

Siirtymä:

Yhteydet eri toimintojen/ikkunoiden/näkymien välillä.

Skenaario:

Kuvaus järjestelmän käytöstä eri näkökulmista.

Standardi:

Periaate, jota noudatetaan järjestelmää suunniteltaessa.

Syöte:

Käyttäjän tai komponentin ohjelmaan syöttämä tieto.

Syöttöpaneeli:

Vaatimuksen tai käyttötapauksen jonkin kuvaustavan mukaisen kuvauksen syöttämistä tukeva paneeli.

Tiedonsaantikerros:

Osajärjestelmä jonka tarkoitus on abstrahoida tapa jolla järjestelmä tallentaa ja hakee tietoa.

Tietokantariippumattomuus:

Järjestelmää ei ole sidottu tiettyyn tietokantaan, muuta kuin tietokannalta vaadittavien ominaisuuksien puolesta.

Tietotyyppi:

Määrittelee muuttujan tai metodin palauttaman arvon tyyppin esim. integer, char, boolean jne.

Toiminto:

Suunnitteludokumenttiin määritellyt vaatimukset toteutetaan järjestelmän toimintoina.

Tuloste:

Komponentin/ohjelman antama tuloste.

Vaatus:

Tuotettavalle ohjelmistolle asetettava määre, joka kertoo mitä järjestelmällä voi tai ei voi tehdä, tai miten järjestelmän tulee toimia.

Vaatusmäärittely:

Ohjelmistotuotannon työvaihe, jossa tuotettavan ohjelmiston ominaisuudet, määritteet ja rajoitteet kartoitetaan yhdessä asiakkaan kanssa.

Valikko:

Valintalista, josta voidaan valita esim. toimintoja.

Yhteys:

Kahden vaatimuksen ja/tai käyttötapauksen liittyminen toisiinsa tai niiden välinen ristiriita.

Yhteyskaavio:

Kaavio, jossa näkyvät valittujen vaatimusten/käyttötapausten yhteydet sekä niiden roolit.

Yhteystyyppi:

Vaatimusten tai käyttötapausten välisen yhteyden tarkoitus/tehtävä järjestelmässä.

Ympäristömuuttuja:

Käyttöjärjestelmän nimetty muuttuja, jota käytetään tiedon tallennukseen ja ohjelmien väliseen kommunikointiin.

Ylläpitäjä:

Järjestelmän hallinnoija.

2 Järjestelmän kokonaiskuvaus

Vaativa-järjestelmä jakaantuu kolmeen alijärjestelmään: ytimeen, tiedonsaantikerrokseen ja liitännäisjärjestelmään. Jaon tarkoituksena on helpottaa ohjelman ylläpitoa ja tukea hyviä ohjelmointitapoja, kuten tiedon kapselointia ja suunnittelumallien käyttöä.

Järjestelmän ydin huolehtii järjestelmän sisäisestä toiminnasta. Se koostuu MVC-mallin mukaisesti ohjaimesta, mallista ja näkymästä. Ohjain koordinoi järjestelmän toimintaa ja välittää tietoa mallin, näkymän ja tiedonsaantikerroksen välillä. Malliin tallennetaan järjestelmän tietosisältö ja näkymä huolehtii järjestelmän käyttöliittymästä.

Tiedonsaantikerroksen tarkoitus on tarjota ytimelle pääsy tietovarastona käytettävään tietokantaan. Sen kautta järjestelmä pääsee käsiksi tiedon pitkäaikaiseen tallennukseen.

Liitännäisjärjestelmän avulla järjestelmää voidaan lisätä uutta toiminnallisuutta ilman muutoksia ytimeen tai tiedonsaantikerrokseen. Tuettuja liitännäisiä ovat vaatimusten ja käyttötapausten kuvauksien syöttöpaneelit, tuotettavan vaatimusdokumentin runkomallit ja tiedon vienti- ja tuontiitännäiset.

2.1 Toteutus- ja toimintaympäristö

Järjestelmä toteutetaan pääosin Java-ohjelmointikielen versiolla 1.5. lukuunottamatta ylläpitäjän käyttöliittymää, joka toteutetaan PHP:llä. Järjestelmän lop-

putestaus suoritetaan Tietojenkäsittelytieteen laitoksen Linux-ympäristössä. Tiedon tallennukseen käytetään Postgres-tietokantaa, mutta järjestelmä rakennetaan toimimaan millä tahansa JDBC-yhteensopivalla tietokannalla, joka tukee BLOB-tietotyyppiä, automaattisesti arvoaan kasvattavia sekvenssikenttiä ja rivitason lukitusta (Row Level Locking).

Koska käytössä on Java-kieli, joka on suunniteltu käyttöjärjestelmäriippumattomaksi, ovat Vaativa-järjestelmän käyttöjärjestelmälle asettamat vaatimukset varsin vaatimattomat. Käyttöjärjestelmältä tarvittavia ominaisuuksia ovat graafinen käyttöliittymä, Java 1.5:n tuki sekä verkkoyhteys tietokantaan. Lisäksi muilla kuin Windows-ympäristöillä vaaditaan, että järjestelmän ympäristömuuttuja BROWSER osoittaa johonkin www-selaimeen. Tämä tarvitaan ohjetiedostojen näyttämistä ja ylläpitäjän käyttöliittymän käynnistämistä varten.

Järjestelmän kehitysympäristönä käytetään Eclipse-kehitysyökalua, ja käyttöliittymän suunnitteluun sen Visual Editor -käyttöliittymäkehittäjä. Järjestelmän tallennukseen ja versionhallintaan kehityksen aikana käytetään Subversion-versionhallintaohjelmistoa.

3 Arkkitehtuurin kuvaus

Vaativa-järjestelmä on suunniteltu modulaariseksi ohjelmistoksi, jossa osajärjestelmien toiminta on selvästi rajattu. Tämä helpottaa kehitystä, sekä järjestelmän ylläpitoa ja jatkokehitystä.

3.1 Sisäinen arkkitehtuuri

Järjestelmän kehitykseen käytetään hyväksitodettuja suunnittelumalleja (mm. Data Access Object, Abstract Factory ja Singleton -mallit) ja ohjelmointitapoja. Komponenttien välisten riippuvuuksien syntymistä vältetään kommunikoimalla tapahtumien ja tarkasti määriteltyjen rajapintojen kautta.

3.1.1 Tiedonsaantikerros eli Data Access Layer

Tiedonsaantikerroksen tarkoitus on abstrahoida tapa, jolla järjestelmä tallentaa ja hakee tietoa. Kaikki tiedonhaku ja muokkaus suoritetaan tiedonsaantiolioiden (Data Access Object, DAO) kautta, jotka tietävät millä tavalla tieto pitää tallentaa ja ha-

kea. Vaativa-järjestelmän kehitysvaiheessa tehdään tiedonsaantikerros, joka käyttää JDBC-yhteensopivaa tietokantaa (kehitysvaiheessa Postgres). Tietokannalta vaaditaan kykyä rivikohtaiseen lukitukseen (Row Level Locking), BLOBien tallennukseen sekä automaattisesti arvoaan kasvattaviin sekvenssikenttiin.

Jatkokehityksessä tiedonsaantikerroksia voidaan vaihtaa, jos tarvitaan jokin muu tietovarasto kuin tietokanta. Tiedonsaantikerroksen vaihtamiselle ei kuitenkaan tässä vaiheessa rakenneta erityistä tukea.

Tiedonsaantikerrosta käytetään DAOFactory-luokan kautta, joka palauttaa luokkia mm. käyttäjien, vaatimusten ja projektien hallintaan. Esimerkiksi kaikki käyttäjien tietojen muokkaus tapahtuu UserDAO:n kautta. DAO-luokat tarjoavat palveluita, joilla tietoa voi lisätä, hakea, muokata ja poistaa.

3.1.2 Liitännäisjärjestelmä eli Plugin system

Vaativa-järjestelmä tukee ylimääräisen toiminnallisuuden lisäämistä liitännäisluokkien kautta. Liitännäisiä on kolmea tyyppiä: InputPlugin, OutputPlugin ja InputPanelPlugin. InputPlugin-tyyppiset liitännäiset tarjoavat tapoja tuoda isoja määriä tietoa järjestelmän ulkopuolelta. OutputPlugin vastaavasti vie järjestelmän tietoa ulos järjestelmästä, kuten tuottaa vaatimusdokumentin rungon. InputPanelPlugin-tyyppiset liitännäiset tarjoavat mahdollisuuden liittää ohjelmaan uuden tyyppisiä tapoja syöttää tietoa.

3.1.3 Ytimen arkkitehtuuri

Ydin koostuu MVC-mallin mukaisesti kolmesta osajärjestelmästä: ohjaimesta, mallista ja näkymästä. Näkymässä on lisäksi huomioitu syöttöpaneeliliitännäisten käyttö.

3.2 Ulkoinen arkkitehtuuri

Vaativa-järjestelmä koostuu kolmesta selvästä osajärjestelmästä: ytimestä, tiedonsaantikerroksesta ja liitännäisjärjestelmästä. Näiden väliset rajapinnat esitellään luvussa 3.3.

Järjestelmän ydin koostuu MVC-mallin mukaan toteutetuista komponenteista, ja sisältää järjestelmän toimintalogiikan ja käyttöliittymän. Ytimessä on järjestelmän

päätoiminnallisuus, jota liitännäisjärjestelmä ja tiedonsaantikerros täydentävät.

Tiedonsaantikerroksen kautta järjestelmä käyttää tietokantaa tai vastaavaa tiedontallennus järjestelmää. Kehitysvaiheessa käytetään tietokantaa. Toteutettava tiedonsaantikerros pystyy käyttämään useimpia moderneja relaatiotietokantoja, joista kehitysvaiheessa on käytössä Postgres. Tarvittaessa tietovarastoliittymä voidaan korvata. Tällöin voidaan käyttää vaihtoehtoisia tiedontalletustapoja, kuten esimerkiksi paikallista tiedostojärjestelmää tai palvelimia verkon yli erilaisilla tiedonsiirtoprotokollilla.

Liitännäisjärjestelmä on ytimeen liittyvä erillinen osajärjestelmä, jonka kautta on mahdollista laajentaa ytimen toiminnallisuutta ilman, että ytimeen tarvitsee tehdä muutoksia. Liitännäisjärjestelmällä lisätään järjestelmään syöttöpaneeleita, eri tapoja tuottaa vaatimuskirjeen runko ja liitännäisiä, joilla toisista järjestelmistä voidaan tuoda tietoa Vaativa-järjestelmään.

3.3 Rajapinnat

Komponenttien ulospäin tarjoamat rajapinnat määrittävät osajärjestelmien tarjoamat palvelut. Seuraavassa lista tärkeimmistä käytettävistä palveluista. Kaikki metodit ovat näkyvyydeltään julkisia (*public*).

3.3.1 Malli

Mallin kaikki metodit palauttavat viitteitä mallissa säilytettäviin olioihin, ei näiden olioiden kopioihin.

void addRequirement(Requirement req); Lisää mallin projektiin uuden vaatimuksen tai vanhan vaatimuksen uuden revision riippuen siitä onko mallissa jo vaatimus, jolla on sama id kuin req:lla. req:n olioviitteet attachments, entities, milestone, status, manager, lastEditor, authorStakeholder, parent, previousVersion ja requirementType täytyy osoittaa nukkeolioihin, joiden muut kentät ovat nulleja(tai nollia) paitsi id, joka asetetaan oikeaksi. req:n olioviitte descriptionin pitää viittata oikeaan olioon. req:n prioritiesin pitää viitata oikeisiin Prioriteetti-oloihin, joiden stakeholder viitteet viittaavat nukkeolioihin. req:n connections-olioviitteiden pitää osoittaa oikeisiin Yhteys-oloihin, joiden object2-olioviitteet soittavat nukkeolioihin.

Requirement getRequirement(int reqId); Hakee mallin projektista viimeisen revision vaatimuksesta, jonka id on reqId. Vanhemmat revisiot löytyvät linkitetystä listasta, jonka juuri uusin revisio on.

List<Requirement> getAllRequirements(); Hakee mallin projektista kaikki vaatimukset. Vanhemmat revisiot löytyvät linkitetyistä listastoista, joiden juuria uusimmat revisiot ovat.

void addUseCase(UseCase useCase); Lisää malliin projektiin uuden käyttötapauksen, tai vanhan käyttötapauksen uuden revision riippuen siitä onko mallissa jo käyttötapaus, jolla on sama id kuin useCase:lla. useCase:n olioviitteet attachments, lastEditor ja entities pitää osoittaa nukkeolioihin, joiden muut kentät ovat nulleja(tai nollia) paitsi id, joka asetetaan oikeaksi. useCase:n olioviitte descriptionin pitää viittata oikeaan olioon. useCase:n stakeholdersin pitää viittata oikeisiin UseCaseToStakeholder-olihiin, joiden stakeholder viitteet viittaavat nukkeolioihin. useCase:n connections-olioviitteiden pitää osoittaa oikeisiin Yhteys-olioihin, joiden object2-olioviitteet osoittavat nukkeolioihin. Samalla lisätään malliin käyttötapaukseen liittyvät liitteet, kuvaus ja UseCaseToStakeHolder yhteydet.

UseCase getUseCase(int caseId); Hakee mallin projektista viimeisimmän revision käyttötapauksesta, jonka id on caseId. Vanhemmat revisiot löytyvät linkitetystä listasta, jonka juuri uusin revisio on.

List<UseCase> getAllUseCases(); Hakee mallin projektista kaikki käyttötapaukset. Vanhemmat revisiot löytyvät linkitetyistä listastoista, joiden juuria uusimmat revisiot ovat.

void addUser(User user); Lisää mallin projektiin uuden käyttäjän tai korvaa mallin projektissa olevan käyttäjän, jolla on sama id kuin userilla, userilla

User getUser(int userId); Hakee mallin projektista käyttäjän, jonka id on userId.

List<User> getAllUser(); Hakee mallin projektista kaikki käyttäjät.

User getCurrentUser(); Hakee parhaillaan sisäänkirjautuneen käyttäjän.

void addAttachment(Attachment attachment); Lisää liitetiedoston mallin projektiin, tai Korvaa mallin projektissa olevan liitetiedoston, jolla on sama

id kuin attachmentilla, attachmentilla. Liitetiedostoa ei kiinnitetä vielä mihinkään vaatimukseen/käyttötapaukseen.

void removeAttachment(Attachment attachment); Poistaa liitetiedoston projektista. Poisto onnistuu vain, jos liitetiedosto ei ole liitettynä mihinkään vaatimukseen tai käyttötapaukseen.

List<Attachment> getAllAttachments(); Hakee mallin projektista kaikki liitetiedostot.

void addRequirementType(RequirementType requirementType); Lisää mallin projektiin uuden vaatimustyyppin tai korvaa mallin projektissa olevan vaatimustyyppin, jolla on sama id kuin requirementTypellä, requirementTypellä.

RequirementType getRequirementType(int requirementTypeId); Hakee mallin projektista vaatimustyyppin, jonka id on requirementTypeId.

List<RequirementType> getAllRequirementTypes(); Hakee mallin projektista kaikki vaatimustyyppit.

void removeRequirementType(RequirementType requirementType); Poistaa mallin projektista vaatimustyyppin. Poisto onnistuu vain, jos vaatimustyyppi ei ole käytössä yhdessäkään vaatimuksessa.

void addStatus(Status status); Lisää mallin projektiin uuden vaatimuksen tilan tai korvaa mallin projektissa olevan vaatimuksen tilan, jolla on sama id kuin statuksella, statuksella.

Status getStatus(int statusId); Hakee mallin projektista vaatimuksen tilan, jonka id on statusId.

List<Status> getAllStatuses(); Hakee mallin projektista kaikki vaatimusten tilat.

void removeStatus(Status status); Poistaa mallin projektista vaatimuksen tilan. Poisto onnistuu vain, jos vaatimuksen tila ei ole käytössä yhdessäkään vaatimuksessa.

void addStatistics(Statistics statistics); Lisää mallin projektiin uuden tilaston, tai korvaa mallin projektissa olevan tilaston, jolla on sama id kuin statisticsilla, statisticsilla.

Statistics getAllStatistics(); Hakee mallin projektista kaikki tilastot.

void addMilestone(Milestone milestone); Lisää mallin projektiin uuden etapin, tai korvaa mallin projektissa olevan etapin, jolla on sama id kuin milestonella, milestonella.

Milestone getMilestone(int milestoneId); Hakee mallin projektista etapin, jonka id on milestoneId.

List<Milestone> getAllMilestones(); Hakee mallin projektista kaikki etapit.

void addStakeholder(Stakeholder holder); Lisää mallin projektiin uuden sidosryhmän, tai korvaa mallin projektissa olevan sidosryhmän, jolla on sama id kuin holderilla, holderilla.

Stakeholder getStakeholder(int stakeholderId); Hakee mallin projektista sidosryhmän, jonka id on stakeholderId.

List<Stakeholder> getAllStakeholders(); Hakee mallin projektista kaikki sidosryhmät.

void removeStakeholder(Stakeholder holder); Poistaa mallin projektista sidosryhmän. Poisto onnistuu vain, jos sidosryhmällä ei ole yhteyksiä mihinkään.

void setProject(Project project); Asettaa mallin projektiksi projectin.

Project getProject(); Hakee mallin projektin.

void addEntity(Entity entity); Lisää mallin projektiin uuden kohdeolion, tai korvaa mallin projektissa olevan kohdeolion, jolla on sama id kuin entityllä, entityllä.

void getEntity(int entityId); Hakee mallin projektista kohdeolion, jonka id on entityId.

List<Entity> getAllEntities(); Hakee mallin projektista kaikki kohdeoliot.

void removeEntity(Entity entity); Poistaa mallin projektista kohdeolion. Poisto onnistuu vain, jos kohdeolio ei ole yhdenkään käyttötapausten tai vaatimuksen kohde.

void addConnectionType(ConnectionType connectionType); Lisää mallin projektiin uuden yhteystyyppin, tai korvaa mallin projektissa olevan yhteystyyppin, jolla on sama id kuin connectionTypellä, connectionTypellä.

ConnectionType getConnectionType(int connectionTypeId); Hakee mallin projektista yhteystyyppin, jonka id on connectionTypeId.

List<ConnectionType> getAllConnectionTypes(); Hakee mallin projektista kaikki yhteystyyppit.

void removeConnectionType(ConnectionType connectionType); Poistaa mallin projektista yhteystyyppin. Poisto onnistuu vain, jos yhteystyyppi ei ole yhdenkään yhteyden tyyppinä.

void addInputPanel(InputPanel inputPanel); Lisää mallin projektiin uuden syöttöpaneelin, tai korvaa mallin projektissa olevan syöttöpaneelin, jolla on sama id kuin inputPanelilla, inputPanelilla.

InputPanel getInputPanel(int inputPanelId); Hakee mallin projektista syöttöpaneelin, jonka id on inputPanelId.

List<InputPanel> getAllInputPanels(); Hakee mallin projektista kaikki syöttöpaneelit.

void removeInputPanel(InputPanel inputPanel); Poistaa mallin projektista syöttöpaneelin. Poisto onnistuu vain, jos syöttöpaneeli ei ole käytössä yhdessäkään vaatimuksessa tai käyttötapauksessa.

void clearModel(); Tyhjentää koko mallin.

void addDatabase(Database db); Lisää malliin tietokannan tiedot(nimi, osoite...).

List<Database> getAllDatabases(); Hakee mallista kaikkien tietokantojen tiedot.

void removeDatabase(Database db); Poistaa mallista tietokannan tiedot.

3.3.2 Näkymä

void startView(); Käynnistää näkymän ja näyttää kirjautumisdialogin.

void closeView(); Sulkee kaikki näkyvissä olevat ikkunat.

void addViewListener(ViewListener listener); Lisää näkymään tapahtuma-kuuntelijan, jolle ilmoitetaan näkymän loogisista tapahtumista.

void setModel(Model model); Asettaa näkymään viitteen käytettävään malliin.

void showError(String message, int options); Näyttää käyttäjälle virhedialogin.

void refreshView(); Päivittää näkymässä näytettävät tiedot.

3.3.3 Tiedonsaantitaso

UserDAO getUserDAO(int projectId); Noutaa tiedonsaantiolion käyttäjien käsittelyyn. Parametri projectId voi olla myös 0, jolloin DAO ei liity mihinkään projektiin, vaan sillä voidaan noutaa järjestelmän yleisiä käyttäjätietoja.

User authenticateUser(String username, String password); Tarkistaa käyttäjän autentikointitiedot tietokannasta. Palauttaa joko käyttäjän tiedot, tai null jos autentikointi ei onnistunut.

void addUser(User user); Lisää käyttäjän tiedot tietokantaan.

User getUser(int userId); Noutaa käyttäjän tiedot tietokannasta.

List<User> getUsersInProject(); Noutaa listan projektiin kuuluvista normaaleista käyttäjistä.

List<User> getUsersNotInProject(); Noutaa listan normaaleista käyttäjistä, jotka eivät kuulu projektiin.

List<User> getAllUsers(); Noutaa listan kaikista järjestelmän normaaleista käyttäjistä.

void saveUser(User user); Päivittää käyttäjän user tiedot tietokantaan.

void refreshUser(User user); Päivittää käyttäjän tiedot tietokannasta malliin.

ProjectDAO getProjectDAO(int projectId); Noutaa tiedonsaantiolion projektien käsittelyyn.

void addProject(Project project); Lisää projektin tiedot tietokantaan.

Project getProject(int projectId); Noutaa projektin tiedot tietokannasta.

List<Project> getAllProjects(); Noutaa listan tietokantaan tallennetuista projekteista.

void saveProject(Project project); Päivittää projektin tiedot mallista tietokantaan.

void refreshProject(Project project); Päivittää projektin tiedot tietokannasta malliin.

boolean isNewest(Date date); Tarkistaa, onko tietokannassa olevaan projektiin tehty muutoksia date-parametrin osoittaman kellonajan jälkeen.

List<InputPanelPlugin> getAllInputPanelPlugins(); Noutaa listan tietokantaan tallennetuista syöttöpaneeleista.

InputPanelPlugin getInputPanelPlugin(int inputPanelId); Noutaa valitun syöttöpaneelin tiedot tietokannasta.

void addEntity(Entity entity); Lisää tietokantaan uuden kohdeolion.

Entity getEntity(int entityId); Noutaa kohdeolion tiedot tietokannasta.

List<Entity> getAllEntities(); Noutaa listan kaikista projektin kohdeolioista.

void saveEntity(Entity entity); Päivittää kohdeolion tiedot mallista tietokantaan.

void refreshEntity(Entity entity); Päivittää kohdeolion tiedot tietokannasta malliin.

void deleteEntity(Entity entity); Poistaa kohdeolion tiedot tietokannasta.

List<Statistics> getStatistics(Date start, Date end); Noutaa projektin tilastotiedot halutulta aikaväliltä.

List<Statistics> getAllStatistics(); Noutaa tietokannasta listan kaikista projektin tilastotiedoista.

Statistics getTodaysStatistics(); Noutaa tietokannasta kuluvan päivän tilastotiedot.

void addStakeholder(Stakeholder stakeholder); Lisää tietokantaan uuden sidosryhmän.

Stakeholder getStakeholder(int stakeholderId); Noutaa sidosryhmän tiedot tietokannasta.

List<Stakeholder> getAllStakeholders(); Noutaa listan kaikista projektiin kuuluvista sidosryhmistä.

void saveStakeholder(Stakeholder stakeholder); Päivittää sidosryhmän tiedot mallista tietokantaan.

void refreshStakeholder(Stakeholder stakeholder); Päivittää sidosryhmän tiedot tietokannasta malliin.

void deleteStakeholder(Stakeholder stakeholder); Poistaa sidosryhmän tiedot tietokannasta.

RequirementDAO getRequirementDAO(int projectId); Noutaa tiedonsaantiolion vaatimusten käsittelyyn.

void addRequirement(Requirement requirement); Lisää vaatimuksen tiedot tietokantaan.

Requirement getRequirement(int requirementId); Noutaa vaatimuksen tiedot tietokannasta.

List<Requirement> getAllRequirements(); Noutaa listan kaikista projektiin kuuluvista vaatimuksista.

void saveRequirement(Requirement requirement); Päivittää vaatimuksen tiedot mallista tietokantaan.

void refreshRequirement(Requirement requirement); Päivittää vaatimuksen tiedot tietokannasta malliin.

UseCaseDAO getUseCaseDAO(int projectId); Noutaa tiedonsaantiolion käyttötapausten käsittelyyn.

void addUseCase(UseCase useCase); Lisää käyttötapausten tiedot tietokantaan.

Usecase `getUseCase(int useCaseId)`; Noutaa käyttötapauksen tiedot tietokannasta.

List<UseCase> `getAllUseCases()`; Noutaa listan kaikista projektiin kuuluvista käyttötapauksista.

void `saveUseCase(UseCase useCase)`; Päivittää käyttötapauksen tiedot mallista tietokantaan.

void `refreshUseCase(UseCase useCase)`; Päivittää käyttötapauksen tiedot tietokannasta malliin.

List<Connection> `getConnections(Connectable connectable)`; Noutaa listan annettuun vaatimukseen tai käyttötapaukseen liittyvistä yhteyksistä.

3.3.4 Liitännäisjärjestelmä

List<Plugin> `listPlugins()`; Etsii kaikki liitännäishakemistosta löytyvät liitännäiset ja palauttaa listan Plugin-olioita jotka sisältävät liitännäisten tiedot.

Plugin `loadPlugin(String name)`; Lataa liitännäisen nimen perusteella ja palauttaa jonkun konkreettisista liitännäisluokista (InputPlugin, OutputPlugin tai InputPanelPlugin).

void `outputModel(Model model)`; OutputPlugin vie mallin tiedot ulos järjestelmästä kutsuttaessa tätä metodia.

void `inputModel(Model model)`; InputPlugin tuo malliin tietoa ulkopuolisesta tietolähteestä tämän metodin kautta.

void `setModel(Model model)`; Antaa InputPanelPlugin tyyppiselle liitännäiselle viitteen malliin johon liitännäinen voi tallentaa tietoa.

4 Tietokanta

Järjestelmän tietokantana käytetään relaatiotietokantana. Tietokannan taulut ja niiden väliset suhteet on esitetty kuvassa 1. Tietokannassa on kaikkiaan 22 taulua, jotka

on esitelty tarkemmin luvussa 4.1. Tietokantatapahtumat on toteutettu transaktioina, joiden ACID-ominaisuudet (atomicity, consistency, isolation, durability) takaavat tietokannan eheyden säilymisen ja järjestelmän toipumisen virhetilanteissa. Esimerkiksi jos järjestelmä kaatuu laitteistovirheen takia, atomisuus (transaktion suoritamat operaatiot joko suoritetaan kaikki tai niitä ei suoriteta lainkaan) takaa sen, että tietokanta säilyy eheänä. Transaktiot pyritään pitämään mahdollisimman lyhyinä, jolloin mahdolliset transaktioiden odotusajat jäävät mahdollisimman lyhyiksi. Perinteinen ACID-transaktiomalli ei sovellu hyvin pitkäkestoisiin transaktioihin, koska se saattaa pakottaa transaktiot odottamaan liian pitkiä aikoja lukkojen vapautumista.

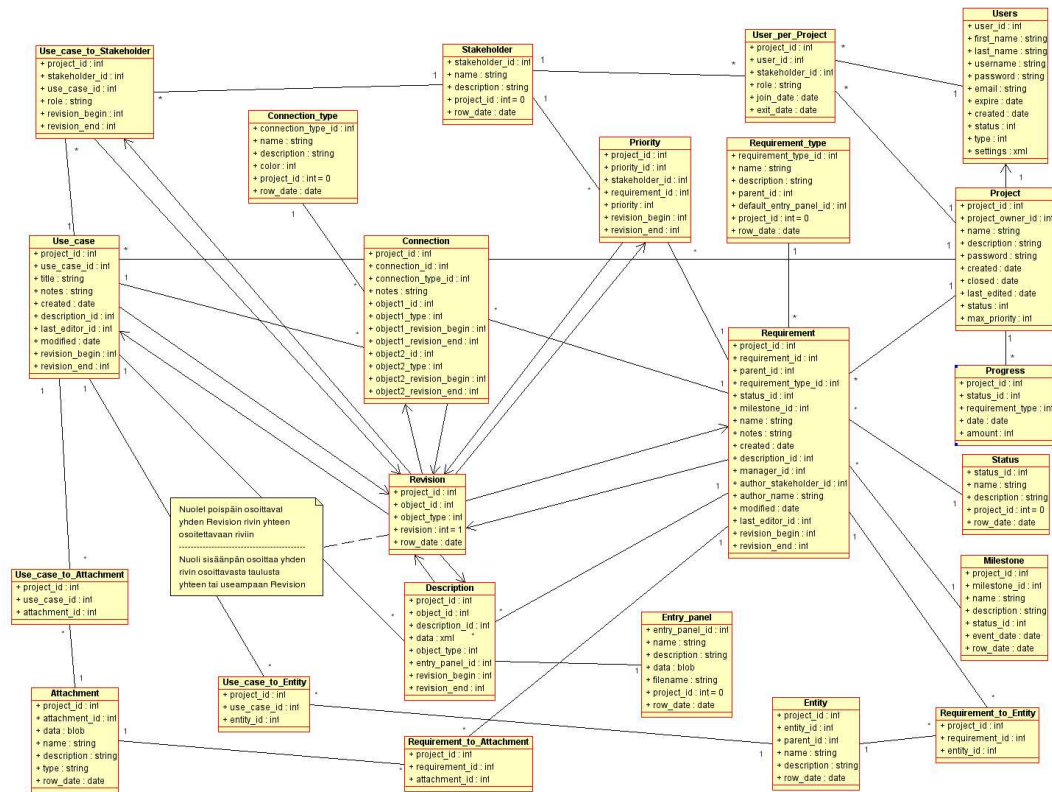
Kaikkien käyttäjän syöttämien tietokantaan tallennettavien tietojen oikeellisuus tarkistetaan järjestelmässä ennen tietojen tallennusta kantaan. Käytännössä transaktiota ei aloiteta eikä siihen sitouduta ennen kuin käyttäjältä saadut tiedot on tarkistettu ja hyväksytty järjestelmän toimesta.

Tietokannan taulut on toteutettu siten, että ne ovat Boyce-Codd -normaalimuodossa. Boyce-Codd -normaalimuoto edellyttää, että minkään monikon mitään attribuuttiarvoa ei voida päätellä taulun muista monikoista tietokantakaavion funktionaalisia riippuvuuksia käyttäen. Ainoat funktionaaliset riippuvuudet ovat avainriippuvuuksia. Lisäksi tietokantakaaviolta edellytetään, että tauluissa ei ole tiedon tarpeetonta toisteisuutta eikä tarpeettomasti tyhjiä arvoja ja tauluissa ei esiinny päivitysanomaliaita.

Tauluissa `connection_type`, `entry_panel`, `requirement_type`, `stakeholder` ja `status` on tietoja, jotka voivat koskea joko jotain tiettyä projektia tai kaikkia projekteja yhteisesti. Tällaiset yhteisesti käytössä olevat tiedot tallennetaan tietokantaan englanniksi, ja niiden käännökset muille kielille tehdään järjestelmän käyttöliittymässä.

Kaikki ensisijaiset avaimet (PRIMARY KEY), jotka muodostuvat vain yhdestä sarakkeesta on toteutettu jaksottaisena (SERIAL, vastaa AUTO_INCREMENT-tyyppiä joissain muissa tietokannoissa) tietotyyppinä. Kun kantaan lisätään alkio saa tämä ensisijaiseksi avaimukseen suurimman ensisijaisen avaimen kasvatettuna yhdellä.

Tietokantaan luodaan alussa projekti ja käyttäjä, jotka saavat tunnukseseen 0. Tämä projekti ja käyttäjä luodaan sen takia, jotta viite-eheys säilyisi yleiskäyttöisissä (project_id = 0) tiedoissa. Kun tietokannasta poistetaan Projekti-taulusta projekti, niin myös kaikki projektin tiedot muissa tauluissa poistuvat.



Kuva 1: Tietokantakaavio

4.1 Tietokantataulut

Tietokantataulut on kuvattu seuraavasti. Aliotsikkona on taulun nimi, jota seuraa sanallinen kuvaus taulusta. Viimeisenä on taulu, jossa on tietokantataulun sarakkeet, sen tietotyyppi ja rajoite, sekä yksilölliset kuvaukset. Kaikissa tauluissa on rajoitteena NOT NULL, paitsi tauluissa joissa on rajoite [NULL].

Sarake:	Kuvaus:
Ensimmäisen sarakkeen nimi: tietotyyppi rajoite	Sanallinen kuvaus ensimmäisestä sarakkeesta
Toisen sarakkeen nimi: tietotyyppi rajoite	Sanallinen kuvaus toisesta sarakkeesta

4.1.1 Attachment

Liitetiedosto-taulussa on käyttötapausten ja vaatimusten sisältämät liitetiedostot. Liitetiedostot ovat käytössä projekteittain. Yksi liitetiedosto voi liittyä useaan käyt-

tötapaukseen ja/tai vaatimukseen ja yhteen vaatimukseen tai liitetiedostoon voi liittyä useita liitetiedostoja. Ensisijainen avain muodostuu `project_id`:stä ja `attachment_id`:stä.

Sarake:	Kuvaus:
<code>project_id</code> : int	Projektin tunnus, jonka olioiden käytössä liitetiedosto on
<code>attachment_id</code> : int	Liitteen projektin sisäinen yksilöllinen tunnus
<code>data</code> : blob	Itse liitetiedoston data
<code>name</code> : varchar[256]	Liitteen nimi
<code>description</code> : text	Liitteen kuvaus
<code>type</code> : varchar[256]	Liitetyypin kuvaus

4.1.2 Connection

Yhteys-taulussa kuvataan kahden olion välistä yhteyttä. Olioiden välillä voi olla monia yhteyksiä. Kun yhteydestä tulee uusi revisio, asetetaan `revision_end`:it osoittamaan samaan olion revisioon, kuin uuden rivin `revision_begin`:it. Tällöin yhteydelle muodostuu olioiden revisioiden välit, joilla yhteys on voimassa. Ensisijainen avain muodostuu `project_id`:stä, `connection_id`:stä ja `revisionista`.

Sarake:	Kuvaus:
project_id: int	Projektin tunnus, jonka kahteen olioon yhteys liittyy
connection_id: int	Yhteyden projektin sisäinen yksilöllinen tunnus
connection_type_id: int	Yhteystyyppin tunnus
notes: text [NULL]	Yhteyteen liittyvät muistiinpanot
object1_id: int	Ensimmäisen olion tunnus
object1_type: int	Ensimmäisen olion tyyppi. 1 = vaatimus, 2 = käyttötapaus
object1_revision_begin: int	Ensimmäisen olion revisio, josta yhteys on voimassa
object1_revision_end: int [NULL]	Ensimmäisen olion revisio, jonka jälkeen yhteys ei ole enää voimassa
object2_id: int	Toisen olion tunnus
object2_type: int	Toisen olion tyyppi. 1 = vaatimus, 2 = käyttötapaus
object2_revision_begin: int	Toisen olion revisio, josta yhteys on voimassa
object2_revision_end: int [NULL]	Toisen olion revisio, jonka jälkeen yhteys ei ole enää voimassa
revision: int	Yhteyden revisio, joka on alussa 1

4.1.3 Connection_type

Yhteystyyppi-taulu kuvaa yhteydelle sen yhteystyyppin. Yhteystyyppi voi olla julkinen (project_id = 0) tai projektikohtainen (project_id = Project.project_id), yhteyksiä voi lisätä uusia. Yhteystyyppejä ovat mm. ristiriita ja yhteys. Ensisijainen avain muodostuu connection_type_id:stä.

Sarake:	Kuvaus:
connection_type_id: int	Yksilöllinen yhteystyyppin tunnus
name: varchar[80]	Yhteystyyppin nimi
description: text	Yhteystyyppin kuvaus
color: int	Yhteystyyppin väri. Väri vaikuttaa käyttöliittymän piirtämien yhteyksien väriin. Tietotyyppiä käsitellään: int[8-15] = punainen, int[8-15] = vihreä, int[0-7] = sininen
project_id: int	Yhteystyyppiin liittyvän projektin mahdollinen tunnus. 0 = nähtävissä kaikilla projekteilla, muuten vastaa projektin id:tä

4.1.4 Description

Kuvaus-taulussa on käyttötapausten ja vaatimusten kuvaukset. Kuvaukset tallennetaan syöttöpaneelin määrittelemässä muodossa. Jokaiseen kuvaukseen liittyy syöttöpaneeli, joka osaa kuvata XML:n luettavassa muodossa. Ensisijainen avain muodostuu project_id:stä, description_id:stä ja revisionista.

Sarake:	Kuvaus:
project_id: int	Projektin tunnus, johon kuvaus kuuluu
description_id: int	Kuvauksen projektin sisäinen yksilöllinen tunnus
data: text	XML-tieto kuvauksen sisällöstä
entry_panel_id: int	Syöttöpaneelin tunnus, jolla sisältö näytetään
revision: int	Kuvauksen revisio, joka on alussa 1

4.1.5 Entity

Kohdeolio-taulussa kuvataan vaatimukseen ja käyttötapauksiin liittyviä kohdeolioita. Kohdeoliolla voi olla vanhempi. Kohdeolioita voivat olla mm. käyttöliittymä, liitännäisjärjestelmä, jokin projektin sidosryhmä tai muu projektiin liittyvä asia. Ensisijainen avain muodostuu project_id:stä ja entity_id:stä.

Sarake:	Kuvaus:
project_id: int	Projektin tunnus, jonka olioihin kohdeolio liittyy
entity_id: int	Kohdeolion projektin sisäinen yksilöllinen tunnus
parent_id: int [NULL]	Kohdeolion vanhemman tunnus
name: varchar[80]	Kohdeolion nimi
description: text	Kohdeolion kuvaus

4.1.6 Entry_panel

Syöttöpaneeli-taulussa ovat järjestelmän syöttöpaneeliliitännäiset. Syöttöpaneelit voivat olla projektikohtaisia (project_id = Project.project_id) tai yleisiä (project_id = 0). Ensisijainen avain muodostuu entry_panel_id:sta.

Sarake:	Kuvaus:
entry_panel_id: int	Syöttöpaneelin yksilöllinen tunnus
name: varchar[80]	Syöttöpaneelin nimi
description: text	Syöttöpaneelin kuvaus
data: blob	Syöttöpaneeli jar-tiedostona
main_class: varchar[256]	Syöttöpaneelin pääluokan nimi
project_id: int	Syöttöpaneeliin liittyvän projektin mahdollinen tunnus. 0 = nähtävissä kaikilla projekteilla, muuten vastaa projektin tunnusta

4.1.7 Milestone

Etappi-taulussa on projektin vaatimuksiin liittyvät etapit. Etapit ovat päivämääriä, jolloin siihen liittyvien vaatimusten täytyy olla status_id:n kuvaamassa tilassa. Ensisijainen avain muodostuu project_id:stä ja milestone_id:stä.

Sarake:	Kuvaus:
project_id: int	Etappiin liittyvän projektin tunnus
status_id: int	Etappiin liittyvän tilan tunnus
milestone_id: int	Etapin yksilöllinen tunnus
name: varchar[80]	Etapin nimi
description: text	Etapin kuvaus
date: date	Etapin päivämäärä

4.1.8 Priority

Prioriteetti-taulussa ovat vaatimukseen liittyvät prioriteetit. Yhdellä vaatimuksella voi olla useita prioriteetteja. Jokaisella prioriteetilla on sen antanut sidosryhmä. Kun prioriteetista tulee uusi revisio, asetetaan req_revision_end osoittamaan saman vaatimuksen uusimpaan revisionumeroon kuin uuden rivin req_revision_begin. Tällöin vaatimukselle muodostuu vaatimuksen revisioiden väli, jolla tämä on voimassa. Ensisijainen avain muodostuu project_id:stä, stakeholder_id:stä, requirement_id:stä ja revisionista.

Sarake:	Kuvaus:
project_id: int	Prioriteettiin liittyvän projektin tunnus
stakeholder_id: int	Prioriteetin antaneen sidosryhmän tunnus
requirement_id: int	Vaatimuksen tunnus, johon prioriteetti liittyy
priority: int	Prioriteetti ($0 \leq \text{prioriteetti} \leq \text{Project.max_priority}$)
req_revision_begin: int	Vaatimuksen revisio, josta lähtien prioriteetti on voimassa
req_revision_end: int [NULL]	Vaatimuksen revisio, jonka jälkeen prioriteetti ei ole enää voimassa
revision: int	Prioriteetin revisio, joka on alussa 1

4.1.9 Project

Projekti-taulussa on kaikki järjestelmän projektien tiedot. Ensisijainen avain muodostuu project_id:stä.

Sarake:	Kuvaus:
project_id: int	Projektin yksilöllinen tunnus
name: varchar[256]	Projektin nimi
description: text	Projektin kuvaus
password: varchar[30] [NULL]	Projektin salasana
created: date	Projektin luontipäivämäärä
closed: date [NULL]	Projektin sulkemispäivämäärä
last_edited: date	Päivämäärä, jolloin projektin jotain tietoa on viimeksi muokattu
status: int	Projektin tila. 0 = avoin, 1 = suljettu, 2 = suljettu, ei voida katsella
max_priority: int	Projektin suurimman prioriteetin arvo (>0)

4.1.10 Requirement

Vaatimustaulussa on järjestelmän vaatimukset projekteittain. Vaatimuksella voi olla vanhempi, jonka sisältö voi toimia vaatimuksen pohjana. Jokaisella vaatimuksella on tyyppi ja tila, jotka tallentuvat tauluihin Requirement_type ja Status. Vaatimukseen voi liittyä etappi, jonka päivämäärään mennessä vaatimuksen tilan pitää olla etapin vaatima tila. Vaatimuksen kuvaus on tallennettu kuvaustauluun. Kuvaus tunnistetaan kuvauksen tunnuksen (description_id) mukaan ja kuvauksen revisio (Description.revision) on description_revision_begin – ja description_revision_end – kenttien arvon välissä. Ensisijainen avain muodostuu project_id:stä, requirement_id:stä ja revisionista.

Sarake:	Kuvaus:
project_id: int	Projektin tunnus, johon vaatimus liittyy
requirement_id: int	Vaatimuksen yksilöllinen tunnus
parent_id: int [NULL]	Vaatimuksen vanhemman tunnus
requirement_type_id: int	Vaatimustyyppin tunniste. Vastaa Requirement_type-taulun tunnistetta
status_id: int	Vaatimuksen tilan tunniste. Vastaa Status-taulun tunnistetta
milestone_id: int [NULL]	Vaatimukseen liittyvän etapin nimi
name: varchar[80]	Vaatimuksen nimi
notes: text [NULL]	Vaatimuksesta kirjoitetut muistiinpanot
created: date	Päivämäärä, jolloin vaatimus on luotu
description_id: int	Kuvauksen tunniste. Vastaa Description-taulun tunnistekenttää
manager_id: int	Vaatimuksen vastuuhenkilön tunniste
author_stakeholder_id: int	Vaatimuksen antaneen sidosryhmän tunnus
author_name: varchar[120]	Sidosryhmästä vaatimuksen antajan nimi
modified: date	Päivämäärä, jolloin vaatimusta on muokattu
last_editor_id: int	Viimeisimmän muokkaajan tunniste
revision: int	Vaatimuksen revisio, joka on alussa 1
description_revision_begin: int	Kuvauksen revisio, josta lähtien kuvaus on voimassa
description_revision_end: int [NULL]	Kuvauksen revisio, jonka jälkeen kuvaus ei ole enää voimassa

4.1.11 Requirement_to_Attachment

Taulu kuvaa monesta moneen suhteen vaatimuksen ja liitteen välille. Yhdellä vaatimuksella voi olla useita liitteitä ja yksi liite voi sisältää useita vaatimuksia.

Sarake:	Kuvaus:
project_id: int	Kuvaukseen liittyvän projektin tunnus
requirement_id: int	Kuvaukseen liittyvän vaatimuksen tunnus
attachment_id: int	Kuvaukseen liittyvän liitteen tunnus

4.1.12 Requirement_to_Entity

Taulu kuvaa monesta moneen suhteen vaatimuksen ja kohdeolion välille. Yhdellä vaatimuksella voi olla useita kohdeolioita ja yhdellä kohdeoliolla voi olla useita vaatimuksia.

Sarake:	Kuvaus:
project_id: int	Kuvaukseen liittyvän projektin tunnus
requirement_id: int	Kuvaukseen liittyvän vaatimuksen tunnus
entity_id: int	Kuvaukseen liittyvän kohdeolion tunnus

4.1.13 Requirement_type

Vaatimustyyppi-aulussa on tallennettuna vaatimukseen liittyvän vaatimustyyppin tiedot. Vaatimustyyppi voi olla vapaasti kaikkien projektien käytössä, jolloin projektin tunnus on 0. Muutoin vaatimustyyppi on sidottu yhteen projektiin ja projektin tunnus vastaa projekti-aulun projektin tunnusta. Ensisijainen avain muodostuu requirement_type_id:stä.

Sarake:	Kuvaus:
requirement_type_id: int	Vaatimustyyppin yksilöllinen tunnus
name: varchar[80]	Vaatimustyyppin nimi
description: text	Vaatimustyyppin kuvaus
parent_id: int [NULL]	Vaatimustyyppin vanhemman tunnus
default_entry_panel_id: int [NULL]	Vaatimustyyppin vakiosyöttöpaneelin tunnus
project_id: int	Vaatimustyyppiin liittyvän projektin tunnus tai 0. Jos vaatimustyyppi on yleiskäyttöinen on tunnus 0. Muuten vaatimustyyppi on projektikohtainen

4.1.14 Stakeholder

Sidosryhmä-aulussa on projektien sidosryhmät. Sidosryhmä voi olla yleisesti käytössä, jolloin projektin tunnus on 0. Muutoin projektin tunnus vastaa projekti-aulun projektin tunnusta ja sidosryhmä on vain tämän projektin käytössä. Ensisijainen avain muodostuu stakeholder_id:stä.

Sarake:	Kuvaus:
stakeholder_id: int	Sidosryhmän yksilöllinen tunnus
name: varchar[80]	Sidosryhmän nimi
description: text	Sidosryhmän kuvaus
project_id: int	Sidosryhmään liittyvän projektin tunnus. 0 jos sidosryhmä on yleiskäyttöinen, eli voidaan käyttää missä vain projektissa. Muuten vastaa projektin tunnusta

4.1.15 Statistics

Tilastot-aulussa on kuvattu projektien tila- ja vaatimustyyppikohtaisten vaatimusten lukumäärä. Tilastot kerätään päivittäin avonaisten projektien tiedoista. Tilastojen avulla voidaan piirtää kuvaaja, jossa nähdään projektikohtaiset vaatimusten määrät ja määrien kehittymiset projektin luonnista lähtien.

Sarake:	Kuvaus:
project_id: int	Projektin tunnus, josta tilastorivi on kerätty
date: date	Päivämäärä, jolloin tieto on kerätty
status_id: int	Vaatimuksen tilan tunnus
requirement_type_id: int	Vaatimustyyppin tunnus
amount: int	Vaatimus-aulun rivien lukumäärä rivejä, joissa on sama requirement_type_id, status_id ja project_id

4.1.16 Status

Tila-aulussa on vaatimuksiin liittyvien tilojen tiedot. Tila voi olla kaikkien projektien yhteiskäytössä, jolloin projektin tunnus on 0 tai se voi liittyä tiettyyn yhteeseen projektiin, jolloin projektin tunnus vastaa projekti-aulun projektin tunnusta. Ensimmäinen avain muodostuu status_id:stä.

Sarake:	Kuvaus:
status_id: int	Tilan yksilöllinen tunnus
name: varchar[80]	Tilan nimi
description: text	Tilan kuvaus
project_id: int	Tilaan liittyvän projektin tunnus. 0 jos tila on yleiskäyttöinen, eli voidaan käyttää missä vain projektissa. Muuten vastaa projektin tunnusta

4.1.17 Use_case

Käyttötapaus-taulussa on järjestelmän käyttötapaukset projekteittain. Käyttötapausten kuvaus on tallennettu kuvaus-tauluun. Kuvaus tunnistetaan kuvauksen tunnuksen (description_id) mukaan ja kuvauksen revisio (Description.revision) on description_revision_begin- ja description_revision_end- kenttien arvon välissä. Ensisijainen avain muodostuu project_id:stä ja use_case_id:stä.

Sarake:	Kuvaus:
project_id: int	Projektin tunnus, johon käyttötapaus kuuluu
use_case_id: int	Käyttötapausten yksilöllinen tunnus
name: varchar[80]	Käyttötapausten nimi
notes: text [NULL]	Käyttötapausten muistiinpanot
created: date	Päivämäärä, jolloin käyttötapaus on luotu
description_id: int	Käyttötapausten kuvauksen tunnus
last_editor_id: int	Viimeisimmän muokkaajan tunnus
revision: int	Käyttötapausten revisio, joka on alussa 1
modified: date	Päivämäärä, jolloin käyttötapausta on muokattu viimeksi
description_revision_begin: int	Kuvauksen revisio, josta lähtien kuvaus on voimassa
description_revision_end: int [NULL]	Kuvauksen revisio, jonka jälkeen kuvaus ei ole enää voimassa

4.1.18 Use_case_to_Attachment

Taulu kuvaa monesta moneen suhteen käyttötapausten ja liitteen välille. Yhdellä käyttötapauksella voi olla useita liitteitä ja yksi liite voi kuulua useisiin käyttötapauksiin.

Sarake:	Kuvaus:
project_id: int	Kuvaukseen liittyvän projektin tunnus
use_case_id: int	Kuvaukseen liittyvän käyttötapauksen tunnus
attachment_id: int	Kuvaukseen liittyvän liitteen tunnus

4.1.19 Use_case_to_Entity

Taulu kuvaa monesta moneen suhteen käyttötapauksen ja kohdeolion välille. Yhdellä käyttötapauksella voi olla useita kohdeolioita ja yhdellä kohdeoliolla voi olla useita käyttötapauksia.

Sarake:	Kuvaus:
project_id: int	Kuvaukseen liittyvän projektin tunnus
use_case_id: int	Kuvaukseen liittyvän käyttötapauksen tunnus
entity_id: int	Kuvaukseen liittyvän kohdeolion tunnus

4.1.20 Use_case_to_Stakeholder

Taulu kuvaa monesta moneen suhteen käyttötapauksen ja sidosryhmän välille. Yhdellä käyttötapauksella voi olla useita sidosryhmiä ja yhdellä sidosryhmällä voi olla useita käyttötapauksia.

Sarake:	Kuvaus:
project_id: int	Kuvaukseen liittyvän projektin tunnus
stakeholder_id: int	Kuvaukseen liittyvän sidosryhmän tunnus
use_case_id: int	Kuvaukseen liittyvän vaatimuksen tunnus
role: varchar[80]	Kuvauksen rooli. Missä roolissa sidosryhmä on käyttötapauksessa

4.1.21 User

Käyttäjä-taulussa on järjestelmän käyttäjien tiedot. Käyttäjät, joiden tyyppi on 0, ovat Vaativa-ohjelman normaalikäyttäjiä. Jos tyyppi on 1, käyttäjä on ylläpitäjä ja hänen tunnuksensa käyvät ylläpitäjän käyttöliittymään. Käyttäjän ohjelman asetukset tallennetaan <settings name="Ohjelman nimi"version="ohjelman asetusten versio"> ja </settings>- tunnusteiden väliin. Tallennettava XML-koodi validoidaan käyttöliittymässä. Tallennettava muoto on:

```
<?xml version="1.0" encoding="UTF-8">
```

```

<vaativa xmlns="...">
  <settings name="..." version="...">
    ...
  </settings>
  <settings name="..." version="...">
    ...
  </settings>
</vaativa>
</xml>

```

Jokainen ohjelman versio lukee tunnistamansa asetukset ja jos sitä ei löydy, niin ohjelma laittaa tiedoston loppuun omat asetuksensa. Ensisijainen avain muodostuu `user_id`:stä.

Sarake:	Kuvaus:
<code>user_id: int</code>	Käyttäjän yksilöllinen tunnus
<code>first_name: varchar[30]</code>	Käyttäjän etunimi
<code>last_name: varchar[60]</code>	Käyttäjän sukunimi
<code>username: varchar[30]</code>	Käyttäjätunnus
<code>password: varchar[80]</code>	Käyttäjän salasana
<code>email: varchar[120]</code>	Käyttäjän sähköpostiosoite
<code>expire: date [NULL]</code>	Päivämäärä, jolloin tunnukset vanhenevat
<code>created: date</code>	Päivämäärä, jolloin tunnus luotiin
<code>status: int</code>	Käyttäjän tila: 0 = avoin, 1 = odottaa ylläpitäjän hyväksyntää, 2 = tunnus jäädytetty ylläpitäjän toimesta
<code>type: int</code>	Käyttäjän tyyppi: 0 = normaali käyttäjä, 1 = ylläpitäjä
<code>settings: text</code>	Käyttäjän asetukset ohjelmaan XML-muodossa

4.1.22 User_per_Project

Käyttäjät projekteittain - taulu kuvaa käyttäjätunnuksen projektiin. Yksi käyttäjä voi olla useassa projektissa ja yhdessä projektissa voi olla useita käyttäjiä. Projektin jäsen voi kuulua tai edustaa jotain sidosryhmää. Ensisijainen avain muodostuu `project_id`:stä ja `user_id`:stä.

Sarake:	Kuvaus:
project_id: int	Projektin tunnus
user_id: int	Käyttäjän tunnus
stakeholder_id: int [NULL]	Käyttäjän sidosryhmän tunnus
role: varchar[120]	Käyttäjän rooli projektissa: owner = omistaja, client = asiakas ja worker = työläinen
join_date: date	Päivämäärä, jolloin käyttäjä liittyi projektiin
exit_date: date [NULL]	Päivämäärä, jolloin käyttäjä lähti projektista

5 Tiedon kulku järjestelmässä

Useimmat järjestelmän tapahtumat alkavat siten, että käyttäjä syöttää tietoa käyttöliittymään. Käyttöliittymä, eli näkymä, välittää tiedot eteenpäin ohjaimelle, joka sitten päättää mitä tulisi tehdä. Ohjain voi tallentaa käyttäjän syöttämiä tietoja tai muita tietoja paikallista lyhytaikaista säilytystä varten malliin tai pitkäaikaista säilytystä varten antaa tiedot DAO:lle, joka huolehtii niiden tallentamisesta tietokantaan. Näkymällä on myös pääsy suoraan malliin ja se voi suorittaa yksinkertaisia operaatioita mallin kanssa ilman ohjainta. DAO toimii siltana ohjaimen ja tietokannan välillä, se siirtää ohjaimen haluamat tiedot tietokantaan ja antaa tietokannasta ohjaimen pyytämät tiedot.

DAO kapseloi tietokannalta saamansa tiedon olioiden sisään, joita muu järjestelmä osaa käyttää. Suurin osa tietokantataulun riveistä on varsin suoraviivaista muuttaa olioiksi, esimerkiksi Requirements-taulun requirement_name sarakkeesta tulee Requirement-olion title-kentän arvo. Suurimmasta osasta tietokannan yhteystauluista ei tehdä olioita, vaan olioiden yhteydet hoituvat olioiden olioviitteiden avulla.

Järjestelmässä on neljä perustapahtumaa: tiedon syöttö, hakeminen, muokkaaminen ja poistaminen.

Perustapauksessa tiedon syötössä käyttäjä syöttää jotain tietoa näkymään, esimerkiksi uuden vaatimuksen. Näkymä välittää tiedot ohjaimelle, joka antaa tiedot DAO:lle siirrettäviksi tietokantaan. DAO tallentaa tiedot tietokantaan ja samalla katsoo onko tietokantaan tehty päivityksiä, joita ei ole vielä ladattu malliin. Lopuksi käyttäjän syöttämät tiedot ja mahdolliset päivitykset tallennetaan malliin.

Tiedon hakeminen tapahtuu useimmiten siten, että käyttäjä antaa näkymälle käskyt tiedon hakemiseksi ja näkymä puolestaan kertoo ohjaimelle, mitä tietoja käyttäjä kaipaa. Ohjain kysyy DAO:lta onko tietokantaan tullut päivityksiä, joita ei ole vielä

ladattu malliin. Jos tietokantaan on tullut päivityksiä, ne haetaan DAO:sta ohjaimen kautta malliin ja sitten ohjain hakee mallista käyttäjän haluamat tiedot näkymälle.

Tiedon muokkaaminen on yhdistelmä tiedon hakemisesta ja syöttämisestä. Ensin muokattava tieto, esimerkiksi vaatimus, haetaan kuten kohdassa tiedon hakeminen, sitten tietoon tehdään halutut muutokset ja lopuksi muuttunut tieto välitetään takaisin säilöön kuten tapauksessa tiedon syöttö sillä erotuksella, että nyt korvataan vanha tieto uudella eikä syötetä täysin uutta.

Järjestelmästä ei poisteta juuri mitään. Suurin osa poistoista on vain näennäisiä poistoja, joissa ainoastaan asetetaan jonkin tilakentän arvo poistetuksi. Tietoa poistetaan järjestelmästä valitsemalla ensin mitä halutaan poistaa, mahdollisesti käyttämällä hyväksi tiedon hakemista. Sitten käyttäjä antaa poistokäskyn näkymälle, joka välittää sen ohjaimelle. Ohjain käskyy mallia poistamaan halutun tiedon ja kertoo DAO:lle, että poistaa tämän tiedon tietokannasta. DAO ja malli tottelevat ja tieto poistetaan järjestelmästä.

Joissain tapauksissa käsiteltävä tieto voi olla jokin liitännäinen, esimerkiksi syöttöpaneeli. Tällöin ohjain käskyttää mallin ja DAO:n sijaan liitännäisjärjestelmää. Tiedon syöttäjän ei aina tarvitse olla käyttäjä - tietoa voidaan hakea myös import ominaisuuden avulla esimerkiksi xml-tiedostoista. Import-liitännäisiä ei toteuteta tämän projektin aikana, mutta niitä voidaan luoda ja liittää järjestelmään myöhemmin.

6 Komponenttien esittely (luokkien ja metodien kuvaus)

6.1 Ydin

6.1.1 Controller

Luokan kuvaus: Controller sisältää järjestelmän toimintalogiikan ja lähes kaiken ohjaukseen kohdistuvan ohjelmakoodin. Controller luokka alustetaan järjestelmän pääluokan toimesta jonka jälkeen Controller luokka aloittaa järjestelmän hallitsemisen.

Metodit

- `public void createModel()` – Luo mallin ja tallentaa ohjaimen viitteen.

- `public void createView()` – Luo näkymän ja tallentaa viitteen.
- `public void initialize()` – Alustaa ohjaimen sisäisen tilan.

6.1.2 DatabaseController

Luokan kuvaus: DatabaseController on Controller-luokan hallinnoima luokka joka keskittyy kuuntelemaan tapahtumia jotka liittyvät tietosisällön hallintaan ulkoisessa varastossa (ts. tietokannassa).

Metodit

- `public void dbEventPerformed(DatabaseEvent databaseEvent)` – Metodia kutsutaan kun halutaan viestittää tietokantatapahtuma. Näitä ovat esimerkiksi tiedon tallennus ja lataus.

6.1.3 ViewController

Luokan kuvaus: Luokka kuuntelee näkymän lähettämiä tapahtumia ja toimii niiden perusteella.

Metodit

- `public void viewEventPerformed(ViewEvent viewEvent)` – Metodia kutsutaan kun näkymä lähettää kaikille kuuntelijoille käyttöliittymään liittyvän tapahtuman. Näitä tapahtumia on esimerkiksi uuden välilehden avaaminen ja järjestelmän sulkeminen.

6.1.4 View

Luokan kuvaus: Näkymän pääluokka joka tarjoaa rajapinnan näkymään.

Näkymän rajapinta on esitetty luvussa 3.3.

Järjestelmän tärkeimmät näkymät sekä osa dialogeista rakennetaan Eclipsen Visual Editor -käyttöliittymäkehittimellä. Niiden luokkarakenne tulee olemaan varsin yksinkertainen, ja generoituu automaattisesti kehittimen määräämään muotoon, johon tehdään jälkeenpäin vain pieniä muokkauksia. Loput käyttöliittymän dialogeista tuotetaan DialogFactory-luokan avulla. Standardidialogeista on kerrottu enemmän luvussa 7.2.

6.1.5 DialogFactory

Luokan kuvaus: Näkymän standardidialogien luomisesta huolehtiva tehdasluokka.

Metodit

- `Map<String, Object> showStandard1Dialog(Map<String, Object> fields)` – Standardin 1 dialogeja luova metodi. Fields-parametrin sisältönä on nimi/arvo -pareja, joista DialogFactory luo dialogiin lisättävät kentät.
- `Map<String, JComponent> showStandard2Dialog(// Map<String, JComponent> fields, List<JButton> buttons)` – Standardin 2 dialogeja luova metodi. Buttons-listan sisältönä on nimi/nappi -pareja.
- `public VaativaTableModel showStandard3Dialog(VaativaTableModel model)` – Standardin 3 dialogeja luova metodi. Parametrinä välitettävä TableModel määrittelee näytettävän taulukon sarakkeet ja sisällön.

Koska käyttöliittymä rakennetaan kehittimillä ja tehdasmetodeilla, ei sen luokkarakennetta haluttu kiinnittää tarkemmin tämän dokumentin tekoon mennessä, vaan se täydennetään tämän dokumentin myöhempiin versioihin toteutuksen edetessä.

6.1.6 Model

Mallin dokumentaatio on tuotettu Javadoc-sivuina, jotka ovat liitteessä 3. Mallin luokkakaavio on esitelty kuvassa 2.

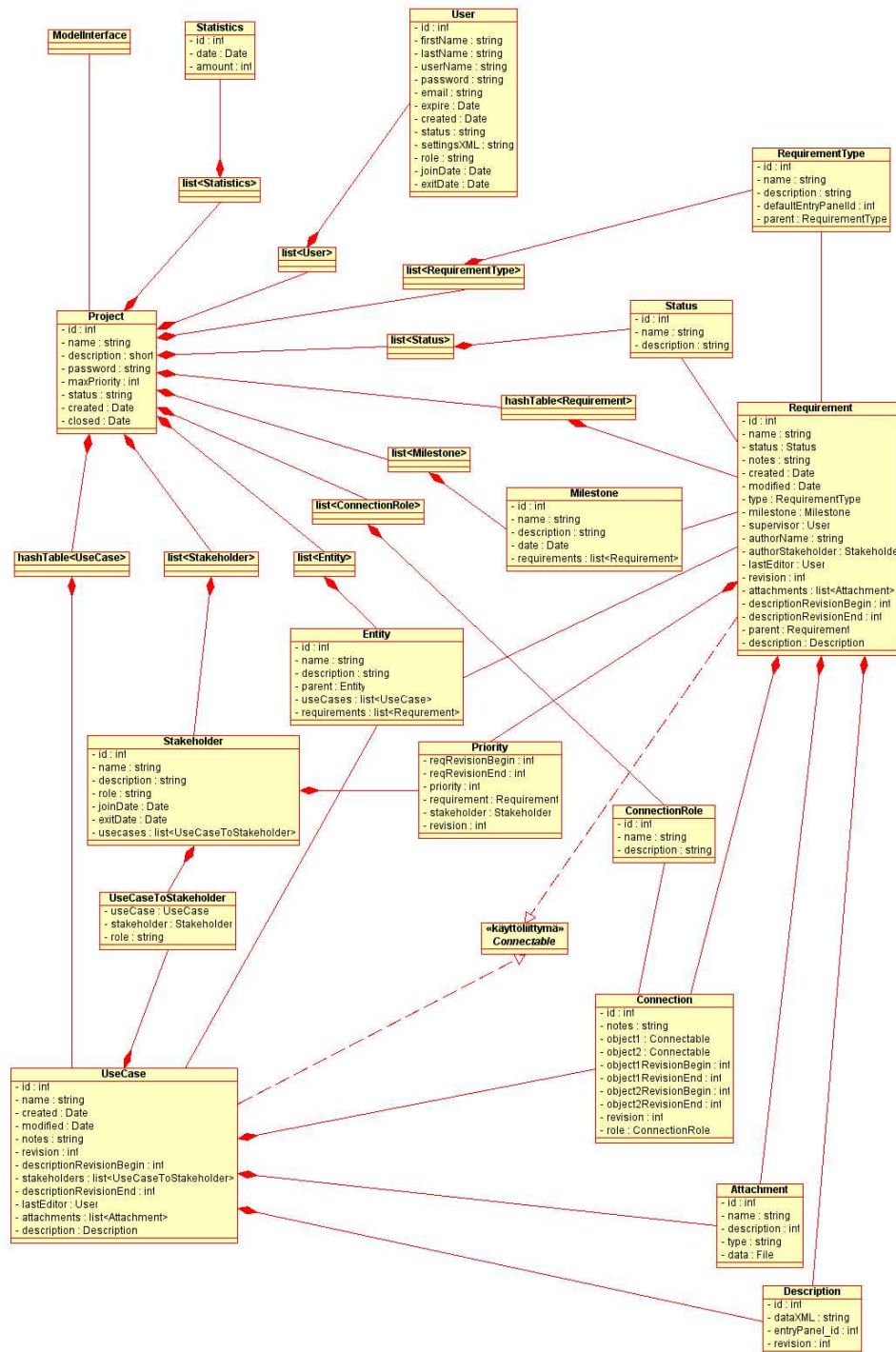
6.2 Tiedonsaantikerros

6.2.1 DAOFactory

Luokan kuvaus: Singleton-ohjelmointimallin mukainen tehdasluokka luo DAO luokkia pyydettyäessä.

Metodit

- `public static DAOFactory getInstance()` – Palauttaa viitteen ainoaan olemassa olevaan ilmentymään.
- `public UserDAO getUserDAO()` – Palauttaa konkreettisen UserDAO luokan kutsujalle.



Kuva 2: Mallin luokkakaavio

- public ProjectDAO getProjectDAO() – Palauttaa konkreettisen ProjectDAO luokan kutsujalle.

- `public UsecaseDAO getUsecaseDAO()` – Palauttaa konkreettisen `UsecaseDAO` luokan kutsujalle.
- `public RequirementDAO getRequirementDAO()` – Palauttaa konkreettisen `RequirementDAO` luokan kutsujalle.

6.3 Liitännäisjärjestelmä

6.3.1 PluginManager

Luokan kuvaus: Hallinnoi liitännäisiä ja lataa niitä tarvittaessa.

Metodit

- `public List listPlugins()` – Palauttaa listan kaikista löydettävissä olevista liitännäisistä. Lista sisältää olioita jotka kuvaavat järjestelmään asennettuja liitännäisiä.
- `public Plugin loadPlugin(String name)` – Hakee liitännäisen nimen perusteella. Liitännäinen on tämän jälkeen valmis käytettäväksi, käyttötapa riippuu liitännäisen tyypistä.

6.3.2 Plugin (rajapintaluokka)

Luokan kuvaus: Kaikki liitännäiset toteuttavat tämän rajapinnan, se määrittelee että luokka on liitännäinen.

Metodit

- `public String getName()` – Palauttaa liitännäisen nimen joka voidaan näyttää käyttäjälle.
- `public String getDescription()` – Palauttaa kuvauksen liitännäisestä.
- **Metodi:** `public int getType()` – Palauttaa tämän liitännäisen tyypin (INPUT, OUTPUT, INPUTPANEL).

6.3.3 InputPlugin (rajapintaluokka extends Plugin)

Luokan kuvaus: Tarkempi rajapinta liitännäisluokille jotka ovat tyyppiä INPUT. Käytetään tuomaan tietoa järjestelmän ulkopuolelta järjestelmän sisäpuolelle.

Metodit

- `public void inputModel(Model model)` – Antaa liitännäiselle viitteen malliin joka voi olla tyhjä tai sisältää tietoa. Liitännäinen lisää malliin tietoa.

6.3.4 OutputPlugin (rajapintaluokka extends Plugin)

Luokan kuvaus: Tarkempi rajapinta liitännäisluokille jotka ovat tyyppiä OUTPUT. Käytetään viemään järjestelmän sisäistä tietoa ulos järjestelmästä. Tällaisia liitännäisiä voisi olla esimerkiksi luokka joka muuttaa tiedon XML muotoon tai tulostettavaan muotoon.

Metodit

- `public void outputModel(Model model)` – Antaa liitännäiselle viitteen malliin. Liitännäinen voi tämän jälkeen viedä mallin tiedot ulos järjestelmästä.

6.3.5 InputPanelPlugin (rajapintaluokka extends Plugin)

Luokan kuvaus: Tarkempi rajapinta liitännäisluokille jotka ovat tyyppiä INPUT-PANEL. Tämän tyyppisiä liitännäisiä voi ladata ohjelman käyttöliittymään ja niiden kautta voidaan syöttää malliin tietoa. Järjestelmässä tulee olemaan oletusarvoisesti liitännäiset vaatimusten ja käyttötapausten syöttöä varten. **Metodit**

- `public void setModel(Model model)` – Antaa liitännäiselle viitteen malliin. Liitännäinen voi tämän jälkeen tallentaa malliin tietoja joita liitännäisen käyttöliittymään syötetään.

7 Käyttöliittymän kuvaus

Vaativa-järjestelmän käyttöliittymä jakautuu kolmeen osaan:

- Sovelluksen aloittaminen
- Käyttäjän käyttöliittymä
- Ylläpitäjän käyttöliittymä

Kaksi ensimmäistä, sovelluksen aloittaminen ja käyttäjän käyttöliittymä, toteutetaan Javan Swing-tekniikalla ja ne kuuluvat tiiviisti yhteen, kun taas ylläpitäjän käyttöliittymä toteutetaan PHP:llä web-sivuina. Sovelluksen aloittaminen on kuitenkin selkeä looginen vaihe ohjelmassa, joten se on kuvattu erillisenä osanaan.

Järjestelmän käyttöliittymän suunnittelussa on pyritty ottamaan huomioon helpokäyttöisyys. Näkymät pyritään otsikoimaan selkeästi ja tarjoamaan käyttäjälle tarvittava ohjeistus siten, että ohjeistus ei kuitenkaan ole häiritsevää. Ergonomia on myös otettava huomioon, kuten hiiren napautusten minimointi ja pikanäppäinkomentojen mahdollisuus tavallisimmissa toiminnoissa. Ohjeistus kuvaa vaatimusten hallinnan työprosessin aina vaatimusten kirjauksesta dokumenttirungon tuottamiseen asti. Ohjeistus tehdään ainakin englanniksi, minkä lisäksi mukaan liitetään tärkeimpien termien käännökset suomeksi.

7.1 Tekniikka

Vaativa-järjestelmän käyttöliittymä toteutetaan pääasiassa Javan Swing-tekniikalla, poikkeuksena PHP-sivuina tehtävä ylläpitäjän käyttöliittymä, josta lisää aliluvussa 7.3.3. Java Swing valittiin tekniikaksi, koska se on projektiryhmälle tuttu. Käyttöliittymän kehitys tehdään kahdella tekniikalla:

1. Järjestelmän tärkeimmät näkymät sekä pieni osa dialogeista rakennetaan Eclipse-IDE:n Visual Editor -käyttöliittymäkehittimellä, minkä jälkeen niiden sisäinen rakenne korjataan selkeämmäksi ja ryhmän standardeja vastaavaksi.
2. Dialogeja varten rakennetaan tehdasluokka, jonka avulla rakennetaan standardityyppisiä dialogi-ikkunoita. Näillä hoidetaan lähes kaikki järjestelmän dialogit. Dialogistandardeista ja tehdasluokasta enemmän aliluvussa 7.2.

7.2 Käyttöliittymän ulkoasustandardit

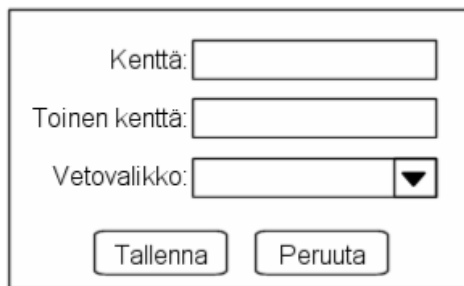
Käyttöliittymän näkymissä pyritään yhtenäiseen ulkoasuun. Välilehdissä ja ikkunoissa noudatetaan yhtenäistä tyyliä. Käyttöliittymän näkymien sisällöt ja ulkoasut selostetaan tarkemmin seuraavissa luvuissa.

Joitakin dialogityyppejä on standardoitu yhtenäisemmän ulkoasun saavuttamiseksi. Jokaiseen näistä tulee alla kuvatun lisäksi järjestelmän tarjoama otsikkorivi, joka

sisältää ainakin sulkunapin, sekä lyhyt ohjeteksti. Järjestelmän sulkunapin painaminen tulkitaan peruutukseksi.

Standardi 1. Nimetyt kentät, 2 nappia

Tämä tyyppi on tarkoitettu yksinkertaisten tiedonkeruudialogien muodoksi. Dialogi luodaan tehdasmetodilla, jolle annetaan lista kentistä ja niiden nimistä, sekä tarvittaessa painikkeiden tekstit. Kentille voidaan antaa myös valmis sisältö, jolloin dialogi toimii muokkausdialogina. Järjestelmä luo dialogin, näyttää sen ja palauttaa sitten olion, joka sisältää lisätyt/muokatut tiedot. Standardi 1 on esitelty kuvassa 3.

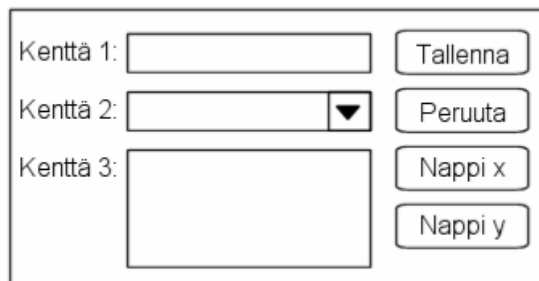


The diagram shows a rectangular dialog box with a white background and a black border. Inside, there are three input fields stacked vertically. The first is labeled 'Kenttä:' and is an empty text box. The second is labeled 'Toinen kenttä:' and is also an empty text box. The third is labeled 'Vetovalikko:' and is a dropdown menu with a small downward-pointing triangle on its right side. Below these fields, there are two buttons: 'Tallenna' on the left and 'Peruuta' on the right.

Kuva 3: Standardi 1

Standardi 2. Nimetyt kentät, enemmän nappeja

Tämä dialogityyppi on tarkoitettu tilanteisiin, joissa tarvitaan enemmän kuin pelkät Tallenna- ja Peruuta-napit. Se luodaan hyvin samalla tapaa kuin edellinenkin, mutta lisäksi annetaan lista napeista ja asetetaan niille tapahtumakuuntelijat. Tämän dialogityypin voi varsin helposti luoda samalla tehdasmetodilla kuin edellisenkin. Standardi 2 on esitelty kuvassa 4.



The diagram shows a rectangular dialog box with a white background and a black border. It contains three input fields on the left, each with a label: 'Kenttä 1:', 'Kenttä 2:', and 'Kenttä 3:'. 'Kenttä 1:' is followed by an empty text box. 'Kenttä 2:' is followed by a dropdown menu with a downward-pointing triangle. 'Kenttä 3:' is followed by a larger empty text box. To the right of these fields, there are four buttons stacked vertically: 'Tallenna', 'Peruuta', 'Nappi x', and 'Nappi y'.

Kuva 4: Standardi 2

Standardi 3. Lista

Tämä tyyppi on tarkoitettu listojen, kuten projektin sidosryhmien tai vaatimustyyppien, muokkaamiseen. Tehdasmetodille annetaan kenttien nimet ja kerrotaan onko taulukon muokkaaminen mahdollista. Jos ei ole, metodi luo dialogin, jossa on pelkkä taulukko. Muuten luodaan myös lisäyskentät sekä Lisää- ja Poista-napit. Kun alla olevasta listasta valitaan rivi, sen tiedot tulevat näkyviin myös ylempiin lisäys/muokkaukenttiin, jossa niitä voidaan muokata. Standardi 3 on esitelty kuvassa 5.

The image shows a dialog box with the following elements:

- Input field labeled "Nimi:"
- Input field labeled "Kuvaus:"
- Buttons: "Lisää", "Muokkaa", "Poista" (located to the right of the input fields)
- A table with 5 rows and 2 columns.
- Buttons: "Tallenna", "Peruuta" (located at the bottom of the dialog)

Kuva 5: Standardi 3

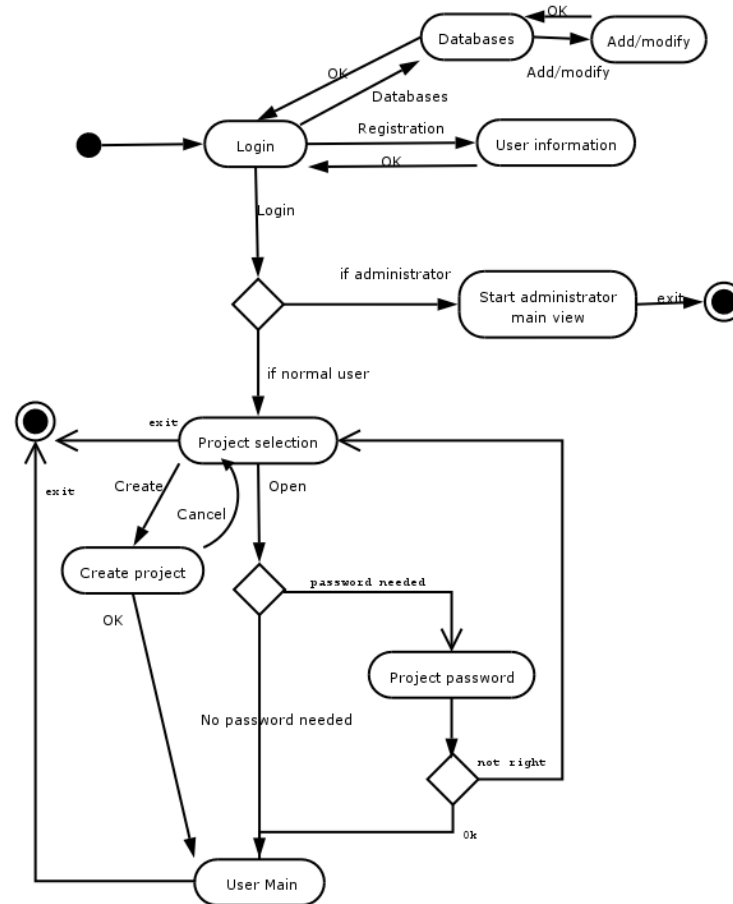
7.3 Käyttöliittymän näkymien kuvaukset

Alla ovat kuvaukset järjestelmän näkymistä. Jokaisesta näkymästä on kerrottu näkymän nimi, tyyppi, kuvaus, näkymän sisältämät tiedot ja kontrollit, ulkoasu (*layout*) sekä siirtymät muihin näkymiin. Koska monet näkymistä ovat suoraviivaisia tiedon muokkaustyökaluja, lisätietoja niiden tiedoista löytyy luvusta 4.1.

7.3.1 Sovelluksen aloittaminen

Uusi käyttäjä valitsee ensin tietokannan, johon liittyy. Vasta tämän jälkeen hän rekisteröityy järjestelmään. Ylläpitäjä hyväksyy rekisteröinnin kuittaamalla käyttäjätietoihin uuden käyttäjätunnuksen. Ylläpitäjä lähettää käyttäjälle sähköpostitse

tiedon rekisteröimisen onnistumisesta. Järjestelmän aloituksen tilakaavio on kuvassa 6.



Kuva 6: Aloituksen tilakaavio

Sisäänkirjautuminen

Tyyppi: Dialogi

Kuvaus: Järjestelmän sisäänkirjautumisruutu. Ensimmäinen asia, jonka käyttäjä näkee. Tähän syötetään käyttäjätunnus ja salasana, valitaan tietokanta johon otetaan yhteys ja painetaan Kirjaudu-painiketta. Tästä pääsee myös rekisteröitymään, sekä lisäämään ja poistamaan tietokantoja. Jos kirjautuva käyttäjä on ylläpitäjä, suljetaan ohjelma ja käynnistetään www-selaimeen ylläpitäjän käyttöliittymä.

Tiedot ja kontrollit:

- Tunnus (tekstikenttä)

- Salasana (salasanakenttä)
- Tietokanta (vetovalikko)
- Kirjaudu (nappi)
- Rekisteröidy... (nappi)
- Tietokannat... (nappi)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Kirjaudu → Projektin valinta (normaalikäyttäjä)
- Kirjaudu → Ylläpitäjän pääsivu (ylläpitäjä)
- Rekisteröidy → Käyttäjän tiedot
- Tietokannat → Tietokannat

Tietokannat

Tyyppi: Dialogi

Kuvaus: Tunnettujen tietokantojen hallintaikkuna. Tässä ikkunassa voi lisätä, muokata ja poistaa tunnettuja tietokantoja. Nämä tiedot tallennetaan paikallisesti, koska tässä vaiheessa ei ole vielä yhteyttä mihinkään tietokantaan. Tästä ikkunasta palatessa Sisäänkirjautuminen-ikkunaan olisi hyvä olla valittuna sama tietokanta, joka oli viimeksi valittuna.

Tiedot ja kontrollit:

- Tietokannat (taulukko)
- Sulje (nappi)
- Lisää... (nappi)
- Muokkaa... (nappi)
- Poista (nappi)

Ulkoasu: Standardi 3

Siirtymät:

- Sulje → Sisäänkirjautuminen
- Lisää → Tietokannan lisäys/muokkaus
- Muokkaa → Tietokannan lisäys/muokkaus

Tietokannan lisäys/muokkaus

Tyyppi: Dialogi

Kuvaus: Dialogi, johon syötetään lisättävän tietokannan tiedot; muokkauksen tapauksessa tiedot ovat valmiina muokattavissa.

Tiedot ja kontrollit:

- Tietokannan nimi (tekstikenttä)
- Tietokannan URL (tekstikenttä)
- Tietokannan ajuri/tyyppi (vetovalikko)
- Tietokannan JDBC-ajuri (vetovalikko)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Standardi 1

Siirtymät:

- Tallenna → Tietokannat
- Peruuta → Tietokannat

Projektin valinta

Tyyppi: Dialogi

Kuvaus: Dialogi, josta käyttäjä valitsee projektin, jota haluaa päästä käsittelemään. Tästä pääsee myös perustamaan uuden projektin tai liittymään johonkin olemassa olevaan projektiin.

Tiedot ja kontrollit:

- Käyttäjän projektit (lista)

- Muut projektit (lista)
- Avaa projekti (nappi)
- Luo uusi projekti... (nappi)
- Liity projektiin (nappi)
- Katsele projektia (nappi)

Ulkoasu: Standardi 2

Siirtymät:

- Avaa projekti → Käyttäjän pääikkuna
- Liity projektiin → Käyttäjän pääikkuna
- Luo projekti → Projektin perustaminen
- Katsele projektia → Käyttäjän pääikkuna

Projektin perustaminen

Tyyppi: Dialogi

Kuvaus: Dialogi, johon käyttäjä syöttää perustettavan projektin tiedot.

Tiedot ja kontrollit:

- Projektin nimi (tekstikenttä)
- Projektin kuvaus (tekstialue)
- Mukaan liitettävät käyttäjät (lista)
- Salasana (salasanakenttä)
- Salasana uudestaan (salasanakenttä)
- Tallenna (nappi)
- Peruuta (nappi)
- Lisää käyttäjiä... (nappi)
- Poista käyttäjä (nappi)

Ulkoasu: Standardi 2

Siirtymät:

- Tallenna → Käyttäjän pääikkuna
- Peruuta → Projektin valinta
- Lisää käyttäjiä → Projektin käyttäjät

7.3.2 Käyttäjän käyttöliittymä

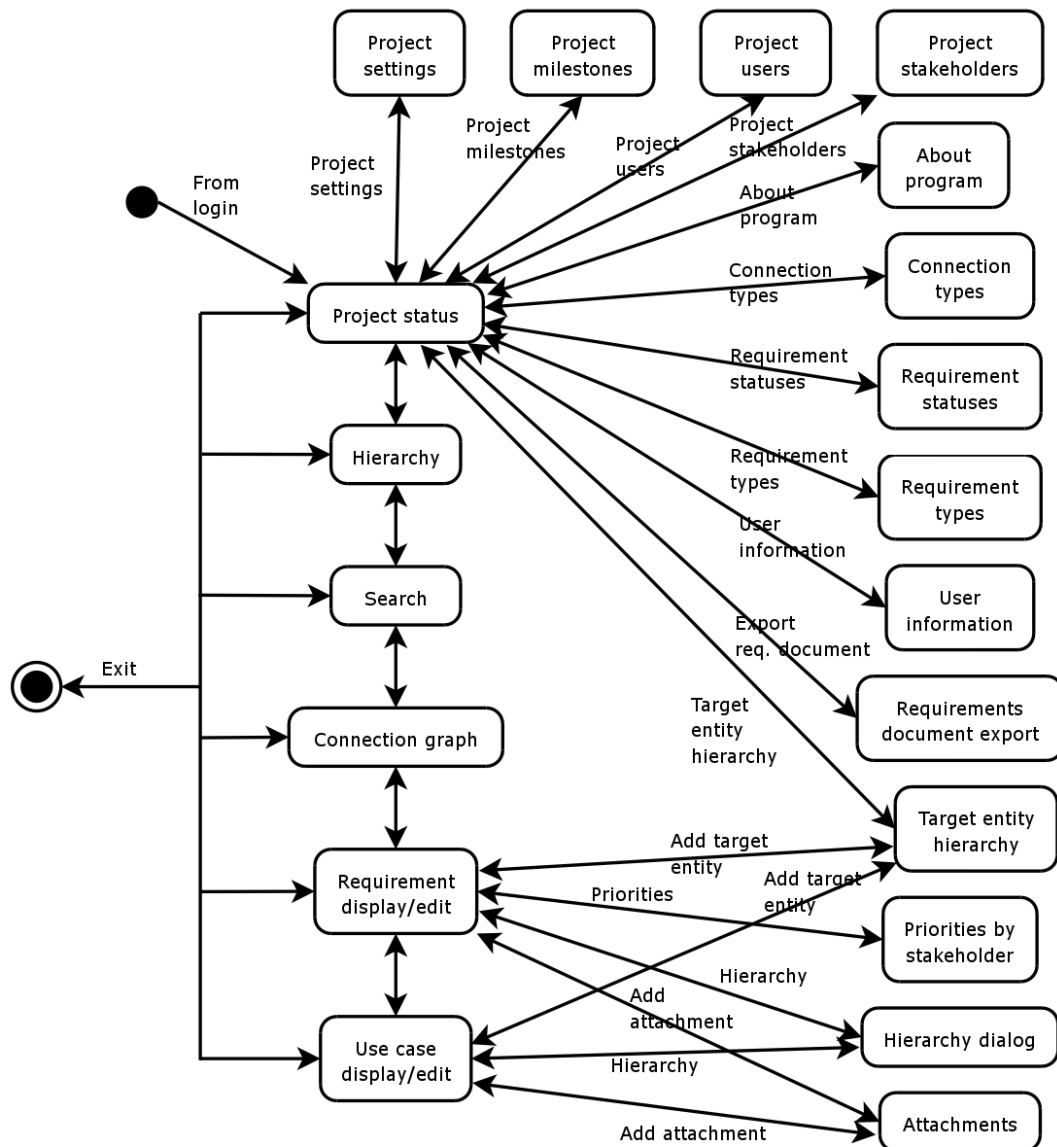
Käyttäjän pääikkunassa tehdään kaikki yhden projektin vaatimushallinnan tärkeät toiminnot (kuva 7). Se koostuu valikkorivistä, työkalurivistä sekä välilehdistä. Välilehdet sisältävät käyttäjän toiminnot. Järjestelmässä on kaksi kiinteää välilehteä: projektin tila, jossa on yleiskuva projektin tiedoista ja työkaluja projektin hallintaan, sekä hierarkia, jossa näkyvät kaikki projektin vaatimukset ja käyttötapaukset hierarkiapuorakenteina. Tarvittaessa avataan vaatimuksen katselu/muokkaus, käyttötapauksen katselu/muokkaus, yhteyskaavio sekä hakutyökalu. Näitä kaikkia voi olla auki useampikin.

Pääikkunan otsikkona on järjestelmän nimi "Vaativa", jonka perässä on käyttäjän ja avoimen projektin nimi. Ikkunan otsikko voisi siis esimerkiksi olla:

Vaativa [Mikko Mallikäyttäjä: Vaativa-projekti]

Käyttäjän käyttöliittymässä on seuraavat välilehdet:

- Projektin tila (pysyvä)
- Hierarkia (pysyvä)
- Haku (tarvittaessa 0-n kpl)
- Vaatimuksen katselu/muokkaus (tarvittaessa 0-n kpl)
- Käyttötapauksen katselu/muokkaus (tarvittaessa 0-n kpl)
- Yhteyskaavio (tarvittaessa 0-n kpl)



Kuva 7: Käyttäjän käyttöliittymän tilakaavio

Ikkunan välilehdet sisältävät muokkauspaneeleita tai katselunäkymiä riippuen kunkin välilehden toiminnoista.

Käyttäjän pääikkuna

Tyyppi: Ikkuna

Kuvaus: Järjestelmän pääikkuna, jossa tehdään kaikki järjestelmän tärkeät toiminnot. Koostuu valikkorivistä, työkalurivistä sekä välilehdistä. Välilehtiä on kah-

denlaisia: kiinteitä ja tarvittaessa avattavia. Kiinteitä välilehtiä ovat Projektin tila, Vaatimushierarkia ja Haku. Tarvittaessa avataan Yhteyskaavio, Vaatimuksen katselu/muokkaus ja Käyttötapausten katselu/muokkaus - näitä kaikkia voi olla auki useampikin. Lisätietoja jokaisen näkymän tiedoissa.

Tiedot ja kontrollit:

- Valikkorivi (kts. Käyttäjän pääikkunan valikot)
- Työkalurivi (kts. Käyttäjän pääikkunan työkalurivi)
- Välilehdet

Ulkoasu: Ylhäältä alas: Valikot, työkalut, välilehdet

Siirtymät:

- Tietokantavirheen sattuessa → Sisäänkirjautuminen
- Välilehden klikkaus → Välilehti

Käyttäjän pääikkunan valikot

Tyyppi: Valikko

Kuvaus: Valikkorivi, josta löytyy suuri osa järjestelmän toiminnoista. Sisältö on koko ajan sama, mutta osa toiminnoista voi olla estetty tilanteen mukaan.

Tiedot ja kontrollit:

- Projekti
 - Uusi vaatimus
 - Uusi käyttötapaus
 - Projektin etapit
 - Projektin käyttäjät
 - Projektin sidosryhmät
 - Kohteet
 - Yhteystyypit
 - Vaatimusten tilat
 - Vaatimusten tyypit

- Projektin asetukset
- Deaktivoi projekti
- Tuo...
- Vie...
- Tuota vaatimusedokumentti
- Kirjautu ulos projektista
- Lopeta
- Työkalut
 - Leikkaa
 - Kopioi
 - Liitä
 - Hakutyökalu
 - Omat tiedot
- Välilehti
 - Aseta muokattavaksi
 - Tallenna muutokset
 - Hylkää muutokset
 - Sulje välilehti
 - Sulje kaikki välilehdet
 - Sulje muut välilehdet
- Ohje
 - Ohjeen aiheet
 - Tietoja Vaativasta

Ulkoasu: Kts. Tiedot ja kontrollit

Siirtymät:

- Projekti: Uusi vaatimus → Vaatimuksen katselu/muokkaus
- Projekti: Uusi käyttötapaus → Käyttötapausten katselu/muokkaus

- Projekti: Projektin etapit → Projektin etapit
- Projekti: Projektin käyttäjät → Projektin käyttäjät
- Projekti: Projektin sidosryhmät → Projektin sidosryhmät
- Projekti: Kohteet → Kohdehierarkia
- Projekti: Yhteystyypit → Yhteystyypit
- Projekti: Vaatimusten tilat → Vaatimusten tilat
- Projekti: Vaatimusten tyypit → Vaatimusten tyypit
- Projekti: Projektin asetukset → Projektin asetukset
- Projekti: Tuo... → Standardi tiedostonvalintadialogi
- Projekti: Vie... → Standardi tiedostonvalintadialogi
- Projekti: Tuota vaatimusdok. → Vaatimusdokumentin tuottaminen
- Projekti: Kirjautu ulos proj. → Projektin valinta
- Projekti: Lopeta → Ohjelman lopetus
- Työkalut: Hakutyökalu → Haku
- Työkalut: Omat tiedot → Käyttäjän tiedot
- Työkalut: Paikalliset asetukset → Paikalliset asetukset
- Ohje: Ohjeen aiheet → Ohjeen etusivu (www-selain)
- Ohje: Tietoja ohjelmasta → Tietoja ohjelmasta

Käyttäjän pääikkunan työkalurivi

Tyyppi: Työkalurivi

Kuvaus: Työkalurivi sijaitsee käyttäjän pääikkunassa valikkorivin alapuolella. Siinä on kerätty järjestelmän tärkeimmät valikkotoiminnot käytön nopeuttamiseksi.

Tiedot ja kontrollit:

- Uusi vaatimus (nappi)

- Uusi käyttötapaus (nappi)
- Aseta muokattavaksi (nappi)
- Tallenna muutokset (nappi)
- Hylkää muutokset (nappi)
- Sulje välilehti (nappi)
- Sulje kaikki välilehdet (nappi)
- Sulje muut välilehdet (nappi)
- Leikkaa (nappi)
- Kopioi (nappi)
- Liitä (nappi)
- Tuota vaatimusdokumentti (nappi)
- Kirjaudu ulos projektista (nappi)
- Lopeta (nappi)
- Ohjeen aiheet (nappi)

Ulkoasu: Vaakasuuntainen rivi nappeja, joissa toimintoja kuvaavat ikonit.

Siirtymät: Kts. Käyttäjän pääikkunan valikot

Projektin tila

Tyyppi: Välilehti (pysyvä)

Kuvaus: Projektin tila -välilehdeltä löytyvät tärkeimmät projektin seurantaan liittyvät tiedot. Näitä tietoja ovat projektin nimi ja kuvaus, projektin käyttäjät, vaatimusten ja käyttötapausten määrät sekä näiden kehittymistä kuvaavat tilastot. Lisäksi välilehdellä on työkaluja projektin tietojen ja asetusten hallintaan.

Tiedot ja kontrollit:

- Projektin nimi (teksti)
- Projektin kuvaus (teksti)

- Projektin käyttäjien määrä (teksti)
- Projektin vaatimusten määrä (teksti)
- Projektin käyttötapausten määrä (teksti)
- Projektin yhteyksien määrä (teksti)
- Projektin sidosryhmien määrä (teksti)
- Projektin etapit (lista)
- Projektin kehitys (kaavio)
- Projektin asetukset (nappi)
- Projektin etapit (nappi)
- Projektin käyttäjät (nappi)
- Projektin sidosryhmät (nappi)
- Kohteet (nappi)
- Yhteysroolit (nappi)
- Vaatimusten tilat (nappi)
- Vaatimusten tyypit (nappi)
- Käyttäjän nimi (teksti)
- Käyttäjän vaatimusten määrä (teksti)
- Käyttäjän käyttötapausten määrä (teksti)
- Omat vaatimukset/käyttötapaukset (nappi)
- Omat tiedot (nappi)
- Uusimmat muokkaukset (lista)
- Eniten ristiriitoja (lista)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Projektin asetukset → Projektin asetukset
- Projektin etapit → Projektin etapit
- Projektin käyttäjät → Projektin käyttäjät
- Projektin sidosryhmät → Projektin sidosryhmät
- Kohteet → Kohdehierarkia
- Yhteysroolit → Yhteystyypit
- Vaatimusten tilat → Vaatimusten tilat
- Vaatimusten tyypit → Vaatimusten tyypit
- Omat vaatimukset/käyttötapaukset → Hakutyökalu
- Käyttäjän tiedot → Käyttäjän tiedot

Hierarkia

Tyyppi: Välilehti (pysyvä)

Kuvaus: Hierarkia-välilehti sisältää kaksi puurakennetta, joissa näkyvät projektin vaatimukset ja käyttötapaukset. Vaatimuspuussa ovat projektin vaatimukset järjestettynä hierarkkisesti, ja jokaisen hierarkiatason sisällä ne ovat otsikon mukaan aakkosjärjestyksessä. Käyttötapaukset ovat puussa sidosryhmittäin, ja jokaisen sidosryhmän alla aakkosjärjestyksessä. Puista voi valita jonkin vaatimuksen tai käyttötapaoksen, jonka voi avata katselu/muokkausnäkömään.

Tiedot ja kontrollit:

- Projektin vaatimukset (puu)
- Projektin käyttötapaukset (puu)

Ulkoasu: Kaksi paneelia, vasemmalla vaatimuspuu, oikealla käyttötapauspauu

Siirtymät:

- Vaatimuksen kaksoisklikkaus → Vaatimuksen muokkaus/katselu
- Käyttötapaoksen kaksoisklikkaus → Käyttötapaoksen muokkaus/katselu

Haku

Tyyppi: Välilehti (avataan tarvittaessa)

Kuvaus: Työkalu, jolla voidaan etsiä haluttuja vaatimuksia ja käyttötapauksia. Tietoa voidaan hakea useiden järjestelmän kenttien perusteella. Käyttäjä syöttää haluamansa hakuehdot hakukenttiin. Hakutyökalun vasemmassa reunassa on joukko hakukenttiä, sekä valintaruudut, joilla valitaan haku vaatimuksista, käyttöliittymistä sekä näiden historiatiedoista. Kun käyttäjä on syöttänyt hakuehdot ja käynnistänyt haun, hakutulokset ilmestyvät taulukkona työkalun oikeaan reunaan.

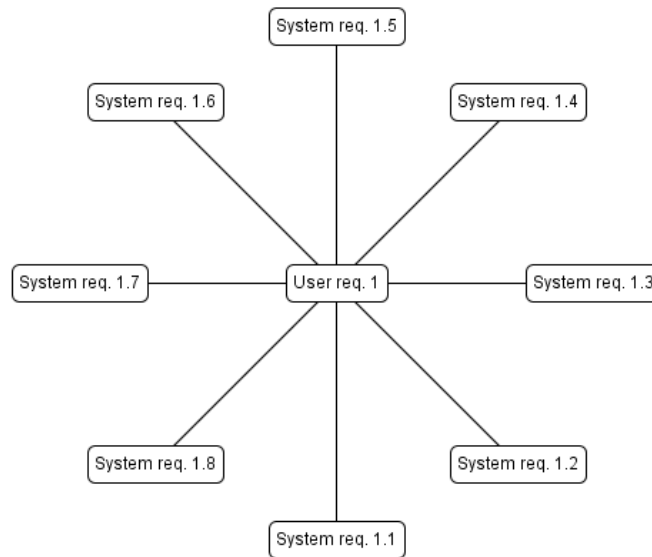
Tiedot ja kontrollit:

- Sisältö (tekstikenttä)
- Otsikko (tekstikenttä)
- Tyyppi (vetovalikko)
- Prioriteetti (tekstikenttä)
- Tila (vetovalikko)
- Sidosryhmä (vetovalikko)
- Etappi (vetovalikko)
- Kohde (vetovalikko)
- Vastuuhenkilö (vetovalikko)
- Muokkaaja (vetovalikko)
- Revision alku (tekstikenttä)
- Revision loppu (tekstikenttä)
- Käynnistä haku (nappi)
- Tyhjennä (nappi)
- Uusi hakutyökalu (nappi)

Ulkoasu: Hakuehdot ja kontrollit vasemmassa reunassa paneelissa, jonka kokoa voi muuttaa. Oikeassa reunassa hakutulokset taulukossa.

Siirtymät:

- Vaatimuksen kaksoisklikkaus → Vaatimuksen muokkaus/katselu
- Käyttötapausten kaksoisklikkaus → Käyttötapausten muokkaus/katselu
- Uusi hakutyökalu → Haku (uusi välilehti)



Kuva 8: Yhteyskaavion malli

Yhteyskaavio

Tyyppi: Välilehti (avataan tarvittaessa)

Kuvaus: Yhteyskaavio liittyy aina johonkin vaatimukseen tai käyttötapaukseen; siinä näkyvät valitun vaatimuksen/käyttötapausten yhteydet sekä niiden roolit. Yhteydet näkyvät kaaviona, jossa valittu vaatimus/käyttötapaus on keskellä ja siitä lähtee viivoja, jotka kuvaavat yhteyksiä (kuva 8). Näiden viivojen toisessa päässä on muita vaatimuksia ja käyttötapausta. Viivan väri kertoo yhteyden tyyppin. Kustakin vaatimuksesta, käyttötapauksesta ja viivasta saa lisätietoja pitämällä hiiren kursoria sen yläpuolella, jolloin kursorin viereen ilmestyy vihjelaatikko (tooltip). Kun jotain muuta vaatimusta/käyttötapausta klikkaa, se siirtyy keskelle ja kaavio siirtyy kuvaamaan sen yhteyksiä. Kaksoisklikkaamalla pääsee katselemaan haluttua vaatimusta/käyttötapausta.

Yhteyskaavio tuotetaan seuraavanlaisella, selkeyden vuoksi yksinkertaistetulla algoritmilla:

`keskus = Yhteyden lähtövaatimus tai -käyttötapaus`

```

yhteydet = Lista yhteyksistä
määrä = yhteydet.size
väli = 360 / määrä
suunta = 0
etäisyys = 100
for yhteydet in yhteydet:
    yhteys.x = x + sin(suunta) * etäisyys
    yhteys.y = y + cos(suunta) * etäisyys
    piirräViiva((keskus.x, keskus.y), (yhteys.x, yhteys.y))
    piirräKohde(yhteys, (yhteys.x, yhteys.y))
    suunta = suunta + väli
piirräKohde(keskus, (keskus.x, keskus.y))

```

Tiedot ja kontrollit:

- Valittu vaatimus/käyttötapaus (pyöristetty suorakulmio)
- Muut vaatimukset/käyttötapaudet (pyöristettyjä suorakulmioita)
- Yhteydet (viivoja)

Ulkoasu: Kts. kuvaus

Siirtymät:

- Vaatimuksen kaksoisklikkaus → Vaatimuksen katselu/muokkaus
- Käyttötapausten kaksoisklikkaus → Käyttötapausten katselu/muokkaus

Vaatimuksen katselu/muokkaus

Tyyppi: Välilehti (avataan tarvittaessa)

Kuvaus: Vaatimuksen katselu- ja muokauslomake. Järjestelmään avataan yksi välilehti katseltavaa/muokattavaa vaatimusta kohti. Välilehti on katselutilassa, jossa tietoja ei voi muuttaa. Kun käyttäjä valitsee muokkaamisen, avataan kentät muokkaukselle. Muokattuaan tietoja käyttäjä valitsee tietojen tallennuksen, jolloin muutuneet tiedot lähetetään tietokantaan ja välilehti palaa katselutilaan. Käyttäjä voi myös hylätä tehdyt muutokset, jolloin tiedot palaavat alkuperäisiksi ja välilehti palaa katselutilaan.

Tiedot ja kontrollit:

- Vaatimuksen otsikko (tekstikenttä)
- Vaatimuksen numero (tekstikenttä)
- Käytettävän syöttöpaneelin valinta (vetovalikko)
- Syöttöpaneeli (syöttöpaneeli)
- Kohteet (lista)
- Lisää kohde (nappi)
- Poista kohde (nappi)
- Vaatimuksen tyyppi (vetovalikko)
- Tila (vetovalikko)
- Prioriteetit (tekstikenttä)
- Etappi (vetovalikko)
- Vastuuhenkilö (vetovalikko)
- Yhteydet (taulukko)
 - Yhteyden kohde
 - Yhteyden rooli (vetovalikko)
- Lisää yhteys (nappi)
- Poista yhteys (nappi)
- Avaa yhteyden kohde (nappi)
- Muistiinpanoja (tekstialue)
- Liitetiedostot (lista)
- Lisää liitetiedosto (nappi)
- Poista liitetiedosto (nappi)
- Tallenna liitetiedosto levyille (nappi)
- Vaatimuksen antaja (tekstikenttä)

- Antajan sidosryhmä (vetovalikko)
- Viimeisin muokkaaaja (tekstikenttä, disabloitu)
- Muokkauspäivämäärä (tekstikenttä, disabloitu)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Vaatimuksen numero (klikkaus) → Hierarkiadialogi (vain vaatimukset)
- Lisää kohde → Kohdehierarkia
- Prioriteetit (klikkaus) → Prioriteetit sidosryhmittäin
- Lisää yhteys → Hierarkiadialogi
- Avaa yhteyden kohde → Vaat./käyttötap. katselu/muokkaus
- Lisää liitetiedosto → Liitetiedostot
- Tallenna liitetiedosto levyille → Standardi tiedostonvalintadialogi

Käyttötapauksen katselu/muokkaus

Tyyppi: Välilehti (avataan tarvittaessa)

Kuvaus: kts. Vaatimuksen katselu/muokkaus.

Tiedot ja kontrollit:

- Käyttötapauksen otsikko (tekstikenttä)
- Käyttötapauksen numero (tekstikenttä, disabloitu)
- Käytettävän syöttöpaneelin valinta (vetovalikko)
- Syöttöpaneeli (syöttöpaneeli)
- Kohteet (lista)
- Lisää kohde (nappi)
- Poista kohde (nappi)
- Sidosryhmä (vetovalikko)

- Vastuuhenkilö (vetovalikko)
- Yhteydet (taulukko)
 - Yhteyden kohde
 - Yhteyden rooli (vetovalikko)
- Lisää yhteys (nappi)
- Poista yhteys (nappi)
- Avaa yhteyden kohde (nappi)
- Muistiinpanoja (tekstialue)
- Liitetiedostot (lista)
- Lisää liitetiedosto (nappi)
- Poista liitetiedosto (nappi)
- Tallenna liitetiedosto levyille (nappi)
- Viimeisin muokkaja (tekstikenttä, disabloitu)
- Muokkauspäivämäärä (tekstikenttä, disabloitu)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Lisää kohde → Kohdehierarkia
- Lisää yhteys → Hierarkiadialogi
- Avaa yhteyden kohde → Vaat./käyttötap. katselu/muokkaus
- Lisää liitetiedosto → Liitetiedostot
- Tallenna liitetiedosto levyille → Standardi tiedostonvalintadialogi

Liitetiedostot

Tyyppi: Dialogi

Kuvaus: Dialogi projektin vaatimuksiin ja käyttötapauksiin tallennettujen liitetiedostojen hallintaan. Tähän tullaan, kun halutaan lisätä vaatimukseen tai käyttötapaukseen liitetiedosto. Tiedostojen lisääminen tehdään Lisää-napilla, jolloin avautuu standardi tiedostonvalinta-dialogi. Taulukosta valittuna oleva tiedosto palautetaan vaatimukseen/käyttötapaukseen Valitse-napilla. Poista-nappi poistaa liitetiedoston koko projektista, mutta tiedosto ei saa olla käytössä missään vaatimuksessa tai käyttötapauksessa.

Tiedot ja kontrollit:

- Tiedostot (taulukko)
 - Tiedoston nimi
 - Tiedoston tyyppi
 - Tiedostoa käyttävien vaatimusten/käyttötapauksien määrä
- Valitse (nappi)
- Peruuta (nappi)
- Lisää... (nappi)
- Poista (nappi)

Ulkoasu: Standardi 2

Siirtymät:

- Valitse → Kutsuva vaatimus/käyttötapaus
- Peruuta → Kutsuva vaatimus/käyttötapaus
- Lisää → Standardi tiedostonvalintadiologi

Hierarkiadiologi

Tyyppi: Dialogi

Kuvaus: Kaksi puurakennetta, joista toisessa on projektin vaatimukset ja toisessa käyttötapaus. Voidaan aktivoida myös siten, että näytetään vain jompikumpi; tätä käytetään hyväksi sijoitettaessa vaatimusta hierarkiaan. Käytetään vaatimuksen sijoituksessa hierarkiaan, sekä kohteen valinnassa. Käyttää hyväksi Hierarkiavälilehden puita.

Tiedot ja kontrollit:

- Vaatimukset (puu)
- Käyttötapaukset (puu)
- Valitse (nappi)
- Peruuta (nappi)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Valitse → Kutsuva vaatimus/käyttötapaus
- Peruuta → Kutsuva vaatimus/käyttötapaus

Kohdehierarkia

Tyyppi: Dialogi

Kuvaus: Dialogi, josta valitaan lisättävä kohde vaatimuksen/käyttötapausten kohdelistaan, sekä lisätään, muokataan, siirrellään ja poistetaan kohteita. Valittavissa olevat kohteet näkyvät puurakenteena, josta voi valita halutun kohteen. Luotaessa uutta kohdetta listaan käyttäjältä kysytään kohta, johon hän haluaa kohteen lisätä. Puurakenne on tasoittain aakkosjärjestyksessä.

Tiedot ja kontrollit:

- Kohteet (puu)
- Valitse (nappi)
- Peruuta (nappi)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Valitse → Kutsuva vaatimus/käyttötapaus
- Peruuta → Kutsuva vaatimus/käyttötapaus

Käyttäjän tiedot

Tyyppi: Dialogi

Kuvaus: Dialogi, jolla käyttäjä rekisteröityy järjestelmän käyttäjäksi, sekä katselee ja muokkaa tietojaan. Sisältää joukon kenttiä, joissa käyttäjän tiedot ovat muokattavassa muodossa.

Tiedot ja kontrollit:

- Nimi (tekstikenttä)
- Salasana (salasanakenttä)
- Salasana uudelleen (salasanakenttä)
- Sähköpostiosoite (tekstikenttä)
- Kielen valinta (vetovalikko)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Standardi 1

Siirtymät:

- Tallenna → Kutsuva näkymä
- Peruuta → Kutsuva näkymä

Projektin asetukset

Tyyppi: Dialogi

Kuvaus: Dialogi, jolla säädetään avoinna olevan projektin asetuksia.

Tiedot ja kontrollit:

- Projektin nimi (tekstikenttä)
- Projektin kuvaus (tekstikenttä)
- Projektin salasana (salasanakenttä)
- Projektin salasana uudestaan (salasanakenttä)
- Prioriteetin alaraja (tekstikenttä)
- Prioriteetin yläaraja (tekstikenttä)

- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Standardi 1

Siirtymät:

- Tallenna → Kutsuva näkymä
- Peruuta → Kutsuva näkymä

Prioriteetit sidosryhmittäin

Tyyppi: Dialogi

Kuvaus: Taulukko, jossa ovat projektin sidosryhmät ja niiden haluamat prioriteetit muokattavalle vaatimukselle.

Tiedot ja kontrollit:

- Sidosryhmät ja prioriteetit (taulukko)
 - Sidosryhmä
 - Prioriteetti (vetovalikko)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Standardi 3

Siirtymät:

- Tallenna → Kutsuva vaatimus
- Peruuta → Kutsuva vaatimus

Vaatimusdokumentin tuottaminen

Tyyppi: Dialogi

Kuvaus: Dialogi, jonka avulla käyttäjä voi tuottaa vaatimusdokumentin rungon. Tässä dialogissa valitaan tuotettavan dokumentin pohja, kieli, mukaan otettavat vaatimukset ja käyttötapaukset sekä dokumenttiin tehtävät otsikot. Otsikoita on ainakin vaatimuksille ja käyttötapauksille omansa.

Tiedot ja kontrollit:

- Käytettävä pohja (vetovalikko)
- Tuotettavan dokumentin kieli (vetovalikko)
- Mukaan otettavat vaatimukset (taulukko)
 - Tila
 - Mukana (valintaruutu)
- Mukaan otettavat käyttötapaukset (taulukko)
 - Sidosryhmät
 - Mukana (valintaruutu)
- Dokumenttiin tehtävät otsikot (lista)
- Lisää otsikko (nappi)
- Poista otsikko (nappi)
- Siirrä otsikkoa ylös (nappi)
- Siirrä otsikkoa alas (nappi)
- Selaa (nappi)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Tallenna → Standardi tiedostonvalintadialogi
- Peruuta → Kutsuva näkymä

Projektin etapit

Tyyppi: Dialogi

Kuvaus: Taulukko projektin etapeista, niiden kuvauksista ja päivämääristä.

Tiedot ja kontrollit:

- Uuden etapin nimi (tekstikenttä)
- Uuden etapin kuvaus (tekstialue)
- Uuden etapin päivämäärä (tekstikenttä)
- Vaadittava tila (vetovalikko)
- Lisää etappi (nappi)
- Etapit (taulukko)
 - Etappi
 - Kuvaus
 - Päivämäärä
- Poista etappi (nappi)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Standardi 3

Siirtymät:

- Tallenna → Kutsuva näkymä
- Peruuta → Kutsuva näkymä

Projektin käyttäjät

Tyyppi: Dialogi

Kuvaus: Projektin käyttäjien hallintatyökalu. Sisältää kaksi käyttäjälistaa, projektiin liitetyt käyttäjät ja muut käyttäjät. Näiden välillä voi siirrellä käyttäjiä joko Lisää- ja Poista-napeilla tai kaksoisklikkaamalla käyttäjän nimeä, jolloin se siirtyy toiseen listaan. Täällä voi myös antaa projektin omistajuuden toiselle käyttäjälle.

Tiedot ja kontrollit:

- Projektin käyttäjät (lista)
- Muut käyttäjät (lista)

- Lisää käyttäjä projektiin (nappi)
- Poista käyttäjä projektista (nappi)
- Tee käyttäjästä projektin omistaja (nappi)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Tallenna → Kutsuva näkymä
- Peruuta → Kutsuva näkymä

Yhteystyypit

Tyyppi: Dialogi

Kuvaus: Dialogi projektin yhteyksien tyyppien hallintaan. Tyypin voi poistaa vain, jos se ei ole käytössä.

Tiedot ja kontrollit:

- Uusi tyyppi (tekstikenttä)
- Uuden tyypin kuvaus (tekstialue)
- Uuden tyypin väri (vetovalikko)
- Lisää tyyppi (nappi)
- Tyypit (taulukko)
 - Tyypin nimi
 - Tyypin kuvaus
 - Tyypin väri
- Poista tyyppi (nappi)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Standardi 3

Siirtymät:

- Tallenna → Kutsuva näkymä
- Peruuta → Kutsuva näkymä

Vaatimusten tilat

Tyyppi: Dialogi

Kuvaus: Dialogi projektin vaatimusten tilojen hallintaan. Tilan voi poistaa vain, jos se ei ole käytössä.

Tiedot ja kontrollit:

- Uusi tila (tekstikenttä)
- Uuden tilan kuvaus (tekstialue)
- Lisää tila (nappi)
- Tilat
 - Tilan nimi
 - Tilan kuvaus
- Poista tila (nappi)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Standardi 3

Siirtymät:

- Tallenna → Kutsuva näkymä
- Peruuta → Kutsuva näkymä

Vaatimusten tyypit

Tyyppi: Dialogi

Kuvaus: Dialogi projektin vaatimusten tyyppien hallintaan. Tyypin voi poistaa vain, jos se ei ole käytössä.

Tiedot ja kontrollit:

- Uusi tyyppi (tekstikenttä)
- Uuden tyypin kuvaus (tekstialue)
- Lisää tyyppi (nappi)
- Tyypit
 - Tyypin nimi
 - Tyypin kuvaus
- Poista tyyppi (nappi)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Standardi 3

Siirtymät:

- Tallenna → Kutsuva näkymä
- Peruuta → Kutsuva näkymä

Projektin sidosryhmät

Tyyppi: Dialogi

Kuvaus: Dialogi projektin sidosryhmien hallintaan. Sidoryhmän voi poistaa vain, jos se ei ole käytössä.

Tiedot ja kontrollit:

- Uusi sidoryhmä (tekstikenttä)
- Uuden sidoryhmän kuvaus (tekstialue)
- Lisää sidoryhmä (nappi)
- Sidoryhmät:

- Sidosryhmän nimi
- Sidosryhmän kuvaus
- Poista sidosryhmä (nappi)
- Tallenna (nappi)
- Peruuta (nappi)

Ulkoasu: Standardi 3

Siirtymät:

- Tallenna → Kutsuva näkymä
- Peruuta → Kutsuva näkymä

Tietoja ohjelmasta

Tyyppi: Dialogi

Kuvaus: Pieni dialogi, joka kertoo tietoja ohjelmasta

Tiedot ja kontrollit:

- Ohjelman nimi
- Versionumero ja päivämäärä
- Tekijät
- Ohjelman kotisivu

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- OK → Kutsuva näkymä

7.3.3 Ylläpitäjän käyttöliittymä

Vaativa-järjestelmän ylläpitäjän käyttöliittymä päätettiin toteuttaa PHP-sivuina. Tällaista toteutusta puolsivat mm. ylläpitäjän roolin erilaisuus normaalin käyttäjän toiminnasta, yhteisten toimien vähyyys, tarve päästä ylläpitäjän toimintoihin käsiksi

paikasta riippumatta sekä ylläpitäjän käyttöliittymän yksinkertaisuus. Myös toisena vaihtoehtona olleelle Java-toteutukselle löytyi monia hyviä argumentteja, minkä takia lopullinen toteutustavan valinta tehtiin kolikkoa heittämällä.

Ylläpitäjän pääsivu

Tyyppi: PHP-sivu

Kuvaus: Ylläpitäjän käyttöliittymän etusivu. Tähän on kerätty yleistietoa järjestelmän tietosisällöstä. Tiedot eivät tällä sivulla ole muokattavassa muodossa.

Tiedot ja kontrollit:

- Käsiteltävän tietokannan osoite
- Projektien määrä järjestelmässä
- Vaatimusten määrä järjestelmässä
- Käyttötapausten määrä järjestelmässä
- Käyttäjien määrä järjestelmässä
- Rekisteröitymistä odottavien käyttäjien määrä järjestelmässä
- Ei-aktiivisten käyttäjien määrä järjestelmässä
- Projektien hallinta (linkki)
- Käyttäjien hallinta (linkki)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Projektien hallinta → Projektien hallinta
- Käyttäjien hallinta → Käyttäjien hallinta

Projektien hallinta

Tyyppi: PHP-sivu

Kuvaus: Sivun järjestelmän projektien hallintaan.

Tiedot ja kontrollit:

- Projektit (lista)

- Projektin nimi (tekstikenttä)
- Projektin kuvaus (tekstialue)
- Projektin vaatimusten määrä
- Projektin käyttötapausten määrä
- Projektin tila (vetovalikko)
- Projektin salasana (salasanakenttä)
- Projektin omistaja (vetovalikko)
- Viimeisin muokkaaja (tekstikenttä, vain luku)
- Projektin käyttäjät (lista)
- Muut käyttäjät (lista)
- Lisää käyttäjä(t) projektiin (nappi)
- Poista käyttäjä(t) projektista (nappi)
- Aktivoi projekti (nappi)
- Deaktivoi projekti (nappi)
- Tallenna muutokset (nappi)
- Tyhjennä (nappi)
- Pääsivu (linkki)
- Käyttäjien hallinta (linkki)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Pääsivu → Ylläpitäjän pääsivu
- Käyttäjien hallinta → Käyttäjien hallinta

Käyttäjien hallinta

Tyyppi: PHP-sivu

Kuvaus: Sivun järjestelmän käyttäjien hallintaan.

Tiedot ja kontrollit:

- Käyttäjät (lista)
- Hyväksyntää odottavat käyttäjät (lista)
- Käyttäjän nimi (tekstikenttä)
- Käyttäjätunnus (tekstikenttä)
- Salasana (salasanakenttä)
- Sähköposti (tekstikenttä)
- Luotu (päivämäärä)
- Käyttöoikeus päättyy (päivämääräkenttä)
- Tila (vetovalikko)
- Tyyppi (vetovalikko)
- Aktivoi käyttäjätunnus (nappi)
- Deaktivoi käyttäjätunnus (nappi)
- Tallenna muutokset (nappi)
- Tyhjennä (nappi)
- Perusta uusi käyttäjä (nappi)
- Pääsivu (linkki)
- Projektien hallinta (linkki)

Ulkoasu: Päätetään toteutusvaiheessa

Siirtymät:

- Pääsivu → Ylläpitäjän pääsivu
- Projektien hallinta → Projektien hallinta

Liite 1. Tuotettava vaatimusdokumentin runko

Järjestelmä tuottaa vaatimusdokumentin sisältörungon. Vaatimusdokumentti sisältää oletuksena luvut johdanto, kokonaiskuva, vaatimukset, käyttötapaukset, toiminnot, käyttöliittymä, ulkoiset liittymät, rajoitukset suunnittelulle ja toteutukselle sekä validointi. Käyttäjää voi halutessaan määritellä otsikot uudelleen, lisätä ja poistaa niitä sekä muuttaa niiden järjestystä. Vain Vaatimukset- ja Käyttötapaukset-luvut sisältävät tietoa; muut ovat pelkkiä otsikoita. Alla on kuvattu oletusotsikoidet suunnitellut sisällöt.

Johdanto-luku sisältää lyhyen kuvauksen dokumentin rakenteesta sekä käytetyistä termeistä ja lyhenteistä. Kokonaiskuva-luvussa kuvataan ongelmakenttää asiakkaan toimialan näkökulmasta. Vaatimukset-luvussa kuvataan järjestelmän toiminnalliset ja laadulliset asiakasvaatimukset. Käyttötapaukset-luvussa kuvataan järjestelmän eri sidosryhmien keskeiset käyttötapaukset ja niiden skenaariot. Toiminnot-luvussa esitellään järjestelmän toiminnot sellaisella tarkkuudella, että niiden perusteella on mahdollista suorittaa hyväksymistestaus. Kuvattujen toimintojen tulee olla siis todennettavissa.

Käyttöliittymä-luvussa esitetään järjestelmän käyttöliittymän ulkoasu. Ulkoiset liittymät -luvussa kuvataan järjestelmän rajapinnat muihin järjestelmiin. Rajoitteet suunnittelulle ja toteutukselle -luvussa kuvataan ne rajoitteet, jotka on otettava huomioon suunnittelussa ja toteutuksessa. Validointi-luvussa osoitetaan, että vaatimusdokumentin toiminnot vastaavat asiakkaan esittämiä vaatimuksia.

Vaatimusdokumentin runko tuotetaan seuraavalla kontekstittomalla kielipöydällä:

LATEX \rightarrow LATEXALKU SISÄLTÖ latexloppu

LATEXALKU \rightarrow alkualku TEKIJÄ alkuloppu

TEKIJÄ \rightarrow tekijä TEKIJÄ λ | tekijä λ

SISÄLTÖ \rightarrow

OTSIKKO VAATIMUKSET OTSIKKO KÄYTTÖTAPAUKSET OTSIKKO |

OTSIKKO KÄYTTÖTAPAUKSET OTSIKKO VAATIMUKSET OTSIKKO

OTSIKKO \rightarrow otsikko OTSIKKO λ | otsikko λ

VAATIMUKSET \rightarrow vaatimusotsikko VAATIMUSTYYPPI

VAATIMUSTYYPPI \rightarrow VAATIMUSTYYPPI VAATIMUSTYYPPI | vaatimustyyppi VAATIMUS | λ

VAATIMUS \rightarrow vaatimus VAATIMUS λ | vaatimus λ

KÄYTTÖTAPAUKSET \rightarrow käyttötapausotsikko SIDOSRYHMÄ

SIDOSRYHMÄ → SIDOSRYHMÄ SIDOSRYHMÄ | sidosryhmä KÄYTTÖTAPAUS | λ
KÄYTTÖTAPAUS → käyttötapaus KÄYTTÖTAPAUS λ | käyttötapaus λ

Suomenkielisessä dokumenttipohjassa päätesymboleiden sisällöt olisivat seuraavat. Englanninkielisessä lähes samanlaiset, mutta jotkin kohdat käännetään englanniksi. Kulmasulkeiden sisällä olevat kohdat ovat merkkauksia, jotka korvataan tarvittavalla sisällöllä.

alkualku:

```
\documentclass[final]{ohtu}
\usepackage{times}
\defprojectgroup{<project_name>}

\defgroupmembers{
```

tekijä:

```
\membername{Olli Opiskelija1}
```

alkuloppu:

```
}

% Projektin asiakkaan nimi.
\defprojectclient{Asiakkaan Nimi}

% Projektiryhmän kotisivu.
\defprojecthomepage{http://www.cs.helsinki.fi/group/esim}

% Projektin johtoryhmä eli vastuuhenkilöt.
\defprojectmasters{
  \mastername{Henkilö1}\\
  \mastername{Henkilö2}
}

\title{Vaatimusdokumentti}
\date{\today}
```

```
\begin{document}
\maketitle
\projectinfopage
\setcounter{page}{1}
\mytableofcontents
```

latexloppu:

```
%\bibliographystyle{tktl}
%\bibliography{lahteet}

\lastpage
\appendices
%\input{liitteet}
\end{document}
```

otsikko:

```
\section{<section_title>}
```

vaatimusotsikko:

```
\section{<section_title>}
```

vaatimustyyppi:

```
\subsection{<requirement_type>}
```

vaatimus: `\textbf{\underline{<requirement_title>}}`

```
\textbf{Tunnus:} <requirement_number>
```

```
\textbf{Kuvaus:}
<requirement_description>
```

```
\textbf{Prioriteetti:} <requirement_priority>
```

```
\textbf{Tila:} <requirement_status>
```

```
\textbf{Etappi:} <requirement_milestone>
```

```
\textbf{Antaja:} <requirement_author>
```


`\textbf{Antajan sidosryhmä:} <requirement_author_stakeholder>`

`\textbf{Vastuuhenkilö:} <requirement_manager>`

`\textbf{Kohteet:}`
`<requirement_targets>`

`\textbf{Yhteydet:}`
`<requirement_connections>`

`% Tiedostot: <requirement_files_list>`

käyttötapausotsikko:

`\section{<section_title>}`

sidosryhmä:

`\subsection{<stakeholder>}`

käyttötapaus: `\textbf{\underline{<usecase_title>}}`

`\textbf{Tunnus:} <usecase_number>`

`\textbf{Kuvaus:}`
`<usecase_description>`

`\textbf{Vastuuhenkilö:} <usecase_manager>`

`\textbf{Kohteet:}`
`<usecase_targets>`

`\textbf{Yhteydet:}`
`<usecase_connections>`

`% Tiedostot: <usecase_files_list>`

Liite 2. Skenaariot

Käyttäjän käyttöliittymäskenaarioita

Käyttäjä rekisteröityy järjestelmään, kirjautuu sisään ja liittyy projektiin

Käyttäjä käynnistää käyttöliittymän. Hän valitsee kirjautumisruudusta tietokannan johon haluaa ottaa yhteyden ja valitsee sen jälkeen rekisteröitymisen. Rekisteröityessään käyttäjä syöttää järjestelmälle nimen, käyttäjätunnuksen, salasanan ja sähköpostiosoitteen. Käyttäjä hyväksyy antamansa tiedot painamalla Register-nappia ja järjestelmä antaa ilmoituksen, jossa kerrotaan että käyttäjän rekisteröityminen on otettu vastaan tai jos käyttäjän antamat tiedot ovat puutteellisia, järjestelmä ilmoittaa siitä. Rekisteröitynyt käyttäjätunnus jää odottamaan projektin omistajan hyväksymistä ja käyttäjän aktivointia.

Kun projektin omistaja on hyväksynyt rekisteröityneen käyttäjän projektiinsa, käyttäjä on valmis kirjautumaan sisään järjestelmään sekä liittymään projektiin/projekteihin. Käyttäjä syöttää käyttäjätunnuksensa ja salasanan ja valitsee tietokantapalvelimen, johon haluaa ottaa yhteyttä. Käyttäjälle avautuu lista niistä projekteista joissa hän on jäsenenä. Hän valitsee projektin jota haluaa käsitellä. Käyttäjälle avautuu ohjelman pääikkuna, jossa näkyvät valitun projektin tiedot.

Käyttäjä lisää viisi vaatimusta ja muokkaa yhtä käyttötapausta

Käyttäjä on kirjautuneena sisään järjestelmään ja tiettyyn projektiin. Hän valitsee Requirement-välilehden ja lisää vaatimukset yksi kerrallaan seuraavasti: käyttäjä syöttää vaaditut tiedot syöttökenttiin mm. vaatimuksen nimen, kuvauksen, tyyppin, tilan, prioriteetin, vaatimuksen kirjaajan nimen sekä käyttäjän, jolle vaatimus suunnataan toteutettavaksi. Käyttäjä vahvistaa tekemänsä vaatimuslisäyksen Tallenna muutokset -napilla. Mikäli jostain pakollisesta syöttökentästä puuttuu tässä vaiheessa tietoa, järjestelmä ilmoittaa asiasta virheilmoituksella.

Muokatakseen käyttötapausta käyttäjä siirtyy Use Case -välilehdelle ja muuttaa, lisää tai poistaa syöttökentissä olevaa tietoa. Käyttäjä hyväksyy tekemänsä muutokset. Mikäli jostain pakollisesta syöttökentästä puuttuu tässä vaiheessa tietoa, järjestelmä ilmoittaa asiasta virheilmoituksella.

Käyttäjä lisää uuden projektin ja oikeutetut käyttäjät

Käyttäjä kirjautuu sisään järjestelmään. Project selection -ruudusta hän valitsee Create new project, jolloin järjestelmä avaa hänelle projektin perustamisruudun. Käyttäjä syöttää vaaditut tiedot syöttökenttiin: projektin nimen ja kuvauksen, projektiin liitettävät käyttäjät ja mahdollisen salasanan (kaksi kertaa). Käyttäjät lisätään Users-kohdasta Add User-painikkeella. Käyttäjät valitaan kaikista järjestelmään rekisteröityneistä käyttäjistä.

Käyttäjä hyväksyy syöttämänsä tiedot. Mikäli jostain pakollisesta syöttökentästä puuttuu tässä vaiheessa tietoa, järjestelmä ilmoittaa asiasta virheilmoituksella. Muussa tapauksessa järjestelmä perustaa uuden projektin ja avaa käyttäjälle ohjelman pääikkunan, jossa näkyvät perustetun projektin tiedot.

Käyttäjä katselee vaatimusten yhteyksiä ja muokkaa niitä

Käyttäjä on kirjautuneen sisään järjestelmään ja tiettyyn projektiin. Hän valitsee jonkin vaatimuksen Connection Graph -välilehden. Käyttäjä valitsee tai etsii haluamansa vaatimukset listalta. Käyttäjä saa nähtäväkseen kaavioesityksen valitsemiensa vaatimusten välisistä yhteyksistä.

Käyttäjä havaitsee virheellisen yhteyden ja haluaa muuttaa sitä tai poistaa sen. Vaatimusten yhteydet toisiin vaatimuksiin määritellään Requirement-välilehdellä kohdassa Connections. Käyttäjä muokkaa vaatimusten yhteyttä ja hyväksyy tekemänsä muutokset. Mikäli jostain pakollisesta syöttökentästä puuttuu tässä vaiheessa tietoa, järjestelmä ilmoittaa asiasta virheilmoituksella.

Käyttäjä valitsee Connection Graph -välilehden ja valitsee uudestaan haluamansa vaatimukset listalta. Käyttäjä saa nähtäväkseen uuden, muokatun kaavioesityksen valitsemiensa vaatimusten välisistä yhteyksistä.

Käyttäjä luo suunnitteludokumentin rungon

Käyttäjä on kirjautuneena järjestelmään ja tiettyyn projektiin. Hän valitsee Project-valikosta suunnitteludokumentin rungon tuottamisen. Käyttäjä valitsee tuottamisen asetukset ja hakemiston, johon dokumentti tuotetaan. Mikäli jostain pakollisesta syöttökentästä puuttuu tässä vaiheessa tietoa, järjestelmä ilmoittaa asiasta virheilmoituksella. Muussa tapauksessa järjestelmä tuottaa halutunlaisen dokumentin rungon.

Projektin omistajan käyttöliittymäskenaarioita

Projektin omistaja muokkaa etappia ja lisää etapin

Projektin omistaja on kirjautuneena järjestelmään ja omistamaansa projektiin. Hän valitsee Project-välilehdeltä etapin ja järjestelmä antaa omistajalle mahdollisuuden muuttaa etapin päivämäärää tai nimeä. Omistaja muuttaa etapin tietoja ja varmistaa tekemänsä muutokset. Mikäli jostain pakollisesta syöttökentästä puuttuu tässä vaiheessa tietoa, järjestelmä ilmoittaa asiasta virheilmoituksella.

Projektin omistaja valitsee etapin lisäämisen. Hän syöttää etapille päivämäärän ja nimen ja varmistaa tekemänsä lisäyksen. Mikäli jostain pakollisesta syöttökentästä puuttuu tässä vaiheessa tietoa, järjestelmä ilmoittaa asiasta virheilmoituksella. Järjestelmä lisää etapin projektiin.

Projektin omistaja sulkee projektin

Projektin omistaja on kirjautuneena järjestelmään ja omistamaansa projektiin. Omistaja vahvistaa projektin sulkemisen Close Project-painikkeella. Järjestelmä asettaa projektin suljettuun tilaan ja omistaja palaa projektin valinta -ruutuun.

Ylläpitäjän käyttöliittymäskenaarioita

Ylläpitäjä poistaa projektin

Ylläpitäjä on kirjautuneena järjestelmään ja on projektien hallintanäkymässä. Ylläpitäjä valitsee projektin/projekteja ja painaa Delete -painiketta. Ylläpitäjä vahvistaa poistamisen, kun järjestelmä vielä varmistaa halutaanko se poistaa. Järjestelmä poistaa valitut projektit ja niiden historiatiedot järjestelmästä kokonaisuudessaan. Mikäli projekti on avoimena poistohetkellä, ylläpitäjälle näytetään virheilmoitus, jossa kerrotaan projektin tilasta ja varmistetaan halutaanko projekti varmasti poistaa. Ylläpitäjä joko hyväksyy tai hylkää projektin poistamisen.

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Attachment

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.Attachment

```
public class Attachment
extends java.lang.Object
```

Liite (Attachment) kapseloi sisäänsä liitetiedoston, joka voi liittyä yhteen tai useampaan vaatimukseen tai käyttötapaukseen. Esimerkiksi käyttötapauksiin liitettävät kuvat kapseloidaan Liitteen sisään. Liite vastaa tietokannan Attachment-taulua.

Constructor Summary

[Attachment](#)()

Konstruktori

[Attachment](#)(int id, java.lang.String name, java.lang.String description, java.lang.String type)

Konstruktori.

Method Summary

java.io.File	getData () Get the value of data
java.lang.String	getDescription () Get the value of description
int	getId () Get the value of id
java.lang.String	getName () Get the value of name
java.lang.String	getType () Get the value of type
void	setData (java.io.File value) Set the value of data
void	setDescription (java.lang.String value) Set the value of description
void	setId (int value) Set the value of id

void	setName (java.lang.String value) Set the value of name
void	setType (java.lang.String value) Set the value of type

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Attachment

```
public Attachment(int id,
                  java.lang.String name,
                  java.lang.String description,
                  java.lang.String type)
```

Konstruktori.

Attachment

```
public Attachment()
```

Konstruktori

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getName

```
public java.lang.String getName()
```

Get the value of name

Returns:

the value of name

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getDescription

```
public java.lang.String getDescription()
```

Get the value of description

Returns:

the value of description

setDescription

```
public void setDescription(java.lang.String value)
```

Set the value of description

getType

```
public java.lang.String getType()
```

Get the value of type

Returns:

the value of type

setType

```
public void setType(java.lang.String value)
```

Set the value of type

getData

```
public java.io.File getData()
```

Get the value of data

Returns:

the value of data

setData

```
public void setData(java.io.File value)
```

Set the value of data

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Interface Connectable

All Known Implementing Classes:

[Requirement](#), [UseCase](#)

```
public interface Connectable
```

Rajapinta Yhdistettävä pääasiallinen käyttö on tehdä Yhteys-luokan toteutuksesta siistimpi. Rajapinnan toteuttavat Käyttötapaus ja Vaatimus

Method Summary

int	getId()
java.lang.String	getName()

Method Detail

getId

```
int getId()
```

getName

```
java.lang.String getName()
```

fi.helsinki.cs.vaativa.model

Class Connection

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.Connection

```
public class Connection
extends java.lang.Object
```

Yhteys-luokka (Connection) kuvaa kahden Yhdistettävän (Connectable) välistä yhteyttä, eli Vaatimusten ja Käyttötapausten mielivaltaisia yhden suhde yhteen yhteyksiä. Yhteydellä on tyyppi, jonka määrittää luokka YhteydenTyyppi (ConnectionType). Yhteys luodaan tietokannan Connection-taulun pohjalta.

Constructor Summary

[Connection](#)()

Konstruktori

[Connection](#)(int id, java.lang.String notes, [Connectable](#) object1, [Connectable](#) object2, int revision, [ConnectionType](#) type)

Konstruktori.

Method Summary

int	getId () Get the value of id
java.lang.String	getNotes () Get the value of notes
Connectable	getObject1 () Get the value of object1
Connectable	getObject2 () Get the value of object2
int	getRevision () Get the value of revision
ConnectionType	getType () Get the value of type
void	setId (int value) Set the value of id
void	setNotes (java.lang.String value) Set the value of notes

void	setObject1 (Connectable value) Set the value of object1
void	setObject2 (Connectable value) Set the value of object2
void	setRevision (int value) Set the value of revision
void	setType (ConnectionType value) Set the value of role

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

Connection

```
public Connection(int id,
                 java.lang.String notes,
                 Connectable object1,
                 Connectable object2,
                 int revision,
                 ConnectionType type)
```

Konstruktori.

Connection

```
public Connection()
```

Konstruktori

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getNotes

```
public java.lang.String getNotes()
```

Get the value of notes

Returns:

the value of notes

setNotes

```
public void setNotes(java.lang.String value)
```

Set the value of notes

getObject1

```
public Connectable getObject1()
```

Get the value of object1

Returns:

the value of object1

setObject1

```
public void setObject1(Connectable value)
```

Set the value of object1

getObject2

```
public Connectable getObject2()
```

Get the value of object2

Returns:

the value of object2

setObject2

```
public void setObject2(Connectable value)
```

Set the value of object2

getRevision

```
public int getRevision()
```

Get the value of revision

Returns:

the value of revision

setRevision

```
public void setRevision(int value)
```

Set the value of revision

getType

```
public ConnectionType getType()
```

Get the value of type

Returns:

the value of type

setType

```
public void setType(ConnectionType value)
```

Set the value of role

[Package](#) **[Class](#)** **[Use Tree](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)**

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class ConnectionType

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.ConnectionType

```
public class ConnectionType
```

```
extends java.lang.Object
```

YhteydenTyyppi (ConnectionRole) kuvaa Yhteyden laatua, eli sitä onko yhteys ristiriita, riippuvuus vai mikä. YhteydenTyyppi luodaan tietokannan Connection_type-taulun perusteella.

Constructor Summary

[ConnectionType](#)()

Konstruktori

[ConnectionType](#)(int id, java.lang.String name, java.lang.String description)

Konstruktori.

Method Summary

java.awt.Color	getColor ()
java.lang.String	getDescription () Get the value of description
int	getId () Get the value of id
java.lang.String	getName () Get the value of name
boolean	isUniversal ()
void	setColor (java.awt.Color color)
void	setDescription (java.lang.String value) Set the value of description
void	setId (int value) Set the value of id
void	setName (java.lang.String value) Set the value of name

void	setUniversal (boolean universal)
------	--

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

ConnectionType

```
public ConnectionType(int id,  
                      java.lang.String name,  
                      java.lang.String description)
```

Konstruktori.

ConnectionType

```
public ConnectionType()
```

Konstruktori

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:
the value of id

setId

```
public void setId(int value)
```

Set the value of id

getName

```
public java.lang.String getName()
```

Get the value of name

Returns:

the value of name

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getDescription

```
public java.lang.String getDescription()
```

Get the value of description

Returns:

the value of description

setDescription

```
public void setDescription(java.lang.String value)
```

Set the value of description

getColor

```
public java.awt.Color getColor()
```

setColor

```
public void setColor(java.awt.Color color)
```

isUniversal

```
public boolean isUniversal()
```

setUniversal

```
public void setUniversal(boolean universal)
```

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Database

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.Database

```
public class Database
extends java.lang.Object
```

Tietokanta-luokka kapseloi sisäänsä yhden tietokannan perustiedot.

Constructor Summary

[Database](#)()

[Database](#)(java.lang.String name, java.lang.String address, int portNumber, java.lang.String description)

Method Summary

java.lang.String	getAddress () Antaa Tietokannan osoitteen.
java.lang.String	getDescription () Antaa Tietokannan kuvauksen.
java.lang.String	getName () Antaa Tietokannan nimen.
int	getPortNumber () Antaa Tietokannan porttinumeron.
void	setAddress (java.lang.String value) Asettaa Tietokannan osoitteen.
void	setDescription (java.lang.String value) Asettaa Tietokannan kuvauksen.
void	setName (java.lang.String value) Asettaa Tietokannan nimen.
void	setPortNumber (int value) Asettaa Tietokannan porttinumeron.

Methods inherited from class java.lang.Object

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
```

Constructor Detail

Database

```
public Database()
```

Database

```
public Database(java.lang.String name,  
                java.lang.String address,  
                int portNumber,  
                java.lang.String description)
```

Method Detail

getName

```
public java.lang.String getName()
```

Antaa Tietokannan nimen.

setName

```
public void setName(java.lang.String value)
```

Asettaa Tietokannan nimen.

getAddress

```
public java.lang.String getAddress()
```

Antaa Tietokannan osoitteen.

setAddress

```
public void setAddress(java.lang.String value)
```

Asettaa Tietokannan osoitteen.

getPortNumber

```
public int getPortNumber()
```

Antaa Tietokannan porttinumeron.

setPortNumber

```
public void setPortNumber(int value)
```

Asettaa Tietokannan porttinumeron.

getDescription

```
public java.lang.String getDescription()
```

Antaa Tietokannan kuvauksen.

setDescription

```
public void setDescription(java.lang.String value)
```

Asettaa Tietokannan kuvauksen.

[Package](#) **[Class](#)** **[Use](#)** **[Tree](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)**

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Description

java.lang.Object

└─ `fi.helsinki.cs.vaativa.model.Description`

```
public class Description  
extends java.lang.Object
```

Kuvaus-luokka kapseloi sisäänsä tiedon Vaatimukseen tai Käyttötapaukseen liittyvästä syöttöpaneelistä ja paneeliin syötetyn tiedon.

Constructor Summary

[Description](#)()

Konstruktori

[Description](#)(int id, java.lang.String dataXML, java.lang.String inputPanelName, int revision)

Konstruktori.

Method Summary

java.lang.String	getDataXML () Get the value of dataXML
int	getId () Get the value of id
java.lang.String	getInputPanelName () Get the value of entryPanelName
int	getRevision () Get the value of revision
void	setDataXML (java.lang.String value) Set the value of dataXML
void	setId (int value) Set the value of id
void	setInputPanelName (java.lang.String value) Set the value of entryPanelName
void	setRevision (int value) Set the value of revision

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Description

```
public Description(int id,  
                    java.lang.String dataXML,  
                    java.lang.String inputPanelName,  
                    int revision)
```

Konstruktori.

Description

```
public Description()
```

Konstruktori

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getDataXML

```
public java.lang.String getDataXML()
```

Get the value of dataXML

Returns:

the value of dataXML

setDataXML

```
public void setDataXML(java.lang.String value)
```

Set the value of dataXML

getInputPanelName

```
public java.lang.String getInputPanelName()
```

Get the value of entryPanelName

Returns:

the value of entryPanelName

setInputPanelName

```
public void setInputPanelName(java.lang.String value)
```

Set the value of entryPanelName

getRevision

```
public int getRevision()
```

Get the value of revision

Returns:

the value of revision

setRevision

```
public void setRevision(int value)
```

Set the value of revision

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Entity

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.Entity

```
public class Entity
extends java.lang.Object
```

KohdeOlio (Entity) kuvaa jotain oliota, johon Vaatimukset tai Käyttötapaukset liittyvät. KohdeOlioita voisivat olla esimerkiksi osajärjestelmät ja ohjelmiston kehityksessä käytettävä prosessi. KohdeOliot voivat muodostaa hierarkkisia puurakenteita. Esimerkiksi osajärjestelmä voi toimia komponenttiansa vanhempana.

Constructor Summary

[Entity](#)()

Konstruktori

[Entity](#)(int id, java.lang.String name, java.lang.String description, [Entity](#) parent, java.util.ArrayList<[UseCase](#)> useCases, java.util.ArrayList<[Requirement](#)> requirements)
Konstruktori.

Method Summary

java.util.ArrayList< Requirement >	getAllRequirements () Get the value of requirements
java.util.ArrayList< UseCase >	getAllUseCases () Get the value of useCases
java.lang.String	getDescription () Get the value of description
int	getId () Get the value of id
java.lang.String	getName () Get the value of name
Entity	getParent () Get the value of parent
void	setDescription (java.lang.String value) Set the value of description
void	setId (int value) Set the value of id

void	setName (java.lang.String value) Set the value of name
void	setParent (Entity value) Set the value of parent
void	setRequirements (java.util.ArrayList< Requirement > value) Set the value of requirements
void	setUseCases (java.util.ArrayList< UseCase > value) Set the value of useCases

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Entity

```
public Entity(int id,
             java.lang.String name,
             java.lang.String description,
             Entity parent,
             java.util.ArrayList<UseCase> useCases,
             java.util.ArrayList<Requirement> requirements)
```

Konstruktori.

Entity

```
public Entity()
```

Konstruktori

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:

the value of id

setId

```
public void setId(int value)
```


Set the value of id

getName

```
public java.lang.String getName()
```

Get the value of name

Returns:

the value of name

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getDescription

```
public java.lang.String getDescription()
```

Get the value of description

Returns:

the value of description

setDescription

```
public void setDescription(java.lang.String value)
```

Set the value of description

getParent

```
public Entity getParent()
```

Get the value of parent

Returns:

the value of parent

setParent

```
public void setParent(Entity value)
```

Set the value of parent

getAllUseCases

```
public java.util.ArrayList<UseCase> getAllUseCases()
```

Get the value of useCases

Returns:

the value of useCases

setUseCases

```
public void setUseCases(java.util.ArrayList<UseCase> value)
```

Set the value of useCases

getAllRequirements

```
public java.util.ArrayList<Requirement> getAllRequirements()
```

Get the value of requirements

Returns:

the value of requirements

setRequirements

```
public void setRequirements(java.util.ArrayList<Requirement> value)
```

Set the value of requirements

[Package](#) **[Class](#)** **[Use Tree](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)**

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class InputPanel

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.InputPanel

```
public class InputPanel
extends java.lang.Object
```

InputPanel-luokka kapseloi sisäänsä syöttöpaneelin hallinnolliset tiedot(nimi, kuvaus, tiedoston nimi).

Constructor Summary

[InputPanel\(\)](#)

Method Summary

java.lang.String	getDescription()
java.lang.String	getFileName()
int	getId()
java.lang.String	getName()
void	setDescription (java.lang.String value)
void	setFileName (java.lang.String value)
void	setId (int value)
void	setName (java.lang.String value)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

InputPanel

```
public InputPanel()
```

Method Detail

getId

```
public int getId()
```

getName

```
public java.lang.String getName()
```

getDescription

```
public java.lang.String getDescription()
```

getFileName

```
public java.lang.String getFileName()
```

setId

```
public void setId(int value)
```

setName

```
public void setName(java.lang.String value)
```

setDescription

```
public void setDescription(java.lang.String value)
```

setFileName

```
public void setFileName(java.lang.String value)
```

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

fi.helsinki.cs.vaativa.model

Class Milestone

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.Milestone

```
public class Milestone
extends java.lang.Object
```

Etappi (Milestone) on lista Vaatimuksista, joiden pitää olla tietyssä Tilassa (Status) tiettyinä päivämääränä.

Constructor Summary

[Milestone](#)()

Luokan parametrin konstruktori.

[Milestone](#)(int id, java.lang.String name, java.lang.String description, java.util.Date date, java.util.ArrayList<[Requirement](#)> requirements)

Luokan konstruktori.

Method Summary

java.util.ArrayList< Requirement >	getAllRequirements () Get the value of requirements
java.util.Date	getDate () Get the value of date
java.lang.String	getDescription () Get the value of description
int	getId () Get the value of id
java.lang.String	getName () Get the value of name
void	setDate (java.util.Date value) Set the value of date
void	setDescription (java.lang.String value) Set the value of description
void	setId (int value) Set the value of id
void	setName (java.lang.String value) Set the value of name

void	setRequirements (java.util.ArrayList< Requirement > value) Set the value of requirements
------	---

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Milestone

```
public Milestone()
```

Luokan parametrin konstruktori.

Milestone

```
public Milestone(int id,  
                 java.lang.String name,  
                 java.lang.String description,  
                 java.util.Date date,  
                 java.util.ArrayList<Requirement> requirements)
```

Luokan konstruktori. Konstruktorin parametrina annetaan luokan muuttujille arvot.

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:
the value of id

setId

```
public void setId(int value)
```

Set the value of id

getName

```
public java.lang.String getName()
```

Get the value of name

Returns:
the value of name

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getDescription

```
public java.lang.String getDescription()
```

Get the value of description

Returns:
the value of description

setDescription

```
public void setDescription(java.lang.String value)
```

Set the value of description

getDate

```
public java.util.Date getDate()
```

Get the value of date

Returns:
the value of date

setDate

```
public void setDate(java.util.Date value)
```

Set the value of date

getAllRequirements

```
public java.util.ArrayList<Requirement> getAllRequirements()
```

Get the value of requirements

Returns:

the value of requirements

setRequirements

```
public void setRequirements(java.util.ArrayList<Requirement> value)
```

Set the value of requirements

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Model

```
java.lang.Object  
└─ fi.helsinki.cs.vaativa.model.Model
```

```
public class Model  
extends java.lang.Object
```

Malli-luokka toimii rajapintana malliin. Yhteen Malliin liittyy aina yksi Projekti.

Constructor Summary

Model ()	Konstruktori.
Model (Project project)	Konstruktori.

Method Summary

void	addAttachment (Attachment attachment) Lisää liitetiedoston mallin projektiin, tai korvaa attachmentilla vanhan liitteen, jolla on sama id kuin attachmentilla.
void	addConnectionType (ConnectionType connectionType) Lisää mallin projektiin uuden yhteystyyppin, tai korvaa connectionTYpellä vanhan yhteystyypin, jolla on sama id kuin connectionTypellä.
void	addDatabase (Database db) Lisää malliin tietokannan tiedot(nimi, osoite...), tai korvaa db:llä vanhan tietokannan tiedot, joilla on sama id kuin db:llä.
void	addEntity (Entity entity) Lisää mallin projektiin uuden kohdeolion, tai korvaa entityllä kohdeolion, jonka id on sama kuin entityllä.
void	addInputPanel (InputPanel inputPanel) Lisää Malliin uuden syöttöpaneelin, tai korvaa inputPanelilla vanhan syöttöpaneelin, jolla on sama id kuin inputPanelilla.
void	addMilestone (Milestone milestone) Lisää mallin projektiin uuden etapin, tai korvaa milestonella vanhan etapin, jolla on sama id kuin milestonella.

void	<p>addRequirement (Requirement req)</p> <p>Lisää mallin projektiin uuden vaatimuksen tai vanhan vaatimuksen uuden revision riippuen siitä onko mallissa jo vaatimus, jolla on sama id kuin req:lla.</p>
void	<p>addRequirementType (RequirementType requirementType)</p> <p>Lisää mallin projektiin uuden vaatimustyyppin, tai korvaa requirementTypellä vanhan vaatimustyyppin, jolla on sama id kuin requirementTypellä.</p>
void	<p>addStakeholder (Stakeholder holder)</p> <p>Lisää mallin projektiin uuden sidosryhmän, tai korvaa holderilla vanhan sidosryhmän, jolla on sama id kuin holderilla.</p>
void	<p>addStatistics (Statistics statistics)</p> <p>Lisää mallin projektiin uuden tilaston, tai korvaa statisticsilla vanhan tilaston, jolla on sama id kuin statisticsilla.</p>
void	<p>addStatus (Status status)</p> <p>Lisää mallin projektiin uuden vaatimuksen tilan, tai korvaa statuksella vanhan tilan, jolla on sama id kuin statuksella.</p>
void	<p>addUseCase (UseCase useCase)</p> <p>Lisää malliin projektiin uuden käyttötapauksen tai vanhan käyttötapauksen uuden revision riippuen siitä onko mallissa jo käyttötapaus, jolla on sama id kuin useCase:lla.</p>
void	<p>addUser (User user)</p> <p>Lisää mallin projektiin uuden käyttäjän, tai korvaa userilla vanhan käyttäjän, jolla on sama id kuin userilla.</p>
java.util.List< Attachment >	<p>getAllAttachments ()</p> <p>Hakee mallin projektista kaikki liitetiedostot.</p>
java.util.List< ConnectionType >	<p>getAllConnectionTypes ()</p> <p>Hakee mallin projektista kaikki yhteystyypit.</p>
java.util.List< Database >	<p>getAllDatabases ()</p> <p>Hakee mallista kaikkien tietokantojen tiedot.</p>
java.util.List< Entity >	<p>getAllEntities ()</p> <p>Hakee mallin projektista kaikki kohdeoliot.</p>
java.util.List< InputPanel >	<p>getAllInputPanels ()</p> <p>Hakee Mallista kaikki Syöttöpaneelit.</p>
java.util.List< Milestone >	<p>getAllMilestones ()</p> <p>Hakee mallin projektista kaikki etapit.</p>
java.util.ArrayList< Requirement >	<p>getAllRequirements ()</p> <p>Hakee mallin projektista kaikki vaatimukset.</p>
java.util.List< RequirementType >	<p>getAllRequirementTypes ()</p> <p>Hakee mallin projektista kaikki vaatimustyypit.</p>
java.util.List< Stakeholder >	<p>getAllStakeholders ()</p> <p>Hakee mallin projektista kaikki sidosryhmät.</p>

java.util.List< Statistics >	getAllStatistics () Hakee mallin projektista kaikki tilastot.
java.util.List< Status >	getAllStatuses () Hakee mallin projektista kaikkien vaatimusten tilat.
java.util.ArrayList< UseCase >	getAllUseCases () Hakee mallin projektista kaikki käyttötapaukset.
java.util.ArrayList< User >	getAllUsers () Hakee mallin projektista kaikki käyttäjät.
Attachment	getAttachment (int attachmentId) Hakee mallin projektista liitteen, jonka id on attachmentId.
ConnectionType	getConnectionType (int connectionTypeId) Hakee mallin projektista yhteystyyppin, jonka id on connectionTypeId.
User	getCurrentUser () Hakee parhaillaan sisäänkirjautuneen käyttäjän.
java.util.Date	getDateOfLastUpdate () Antaa viimeisimmän päivityksen ajan.
Entity	getEntity (int entityId) Hakee mallin projektista kohdeolion, jonka id on entityId.
boolean	getHistoryEnabled () Antaa tiedon siitä, onko historia käytettävissä (true) vai ei (false).
InputPanel	getInputPanel (int inputPanelId) Hakee Mallista Syöttöpaneelin, jonka id on inputPanelId.
Milestone	getMilestone (int milestoneId) Hakee mallin projektista etapin, jonka id on milestoneId.
Project	getProject () Hakee mallin projektin.
Requirement	getRequirement (int reqId) Hakee mallin projektista vaatimuksen, jonka id on reqId.
RequirementType	getRequirementType (int requirementTypeId) Hakee mallin projektista vaatimustyyppin, jonka id on requirementTypeId.
Stakeholder	getStakeholder (int stakeholderId) Hakee mallin projektista sidosryhmän, jonka id on stakeholderId.
Status	getStatus (int statusId) Hakee mallin projektista vaatimuksen tilan, jonka id on statusId.
UseCase	getUseCase (int caseId) Hakee mallin projektista käyttötapauksen, jonka id on caseId.
User	getUser (int userId) Hakee mallin projektista käyttäjän, jonka id on userId.

void	removeAttachment (int attachmentId) Poistaa liitetiedoston projektista.
void	removeConnectionType (ConnectionType connectionType) Poistaa mallin projektista yhteystyyppin.
void	removeDatabase (Database db) Poistaa mallista tietokannan tiedot.
void	removeEntity (Entity entity) Poistaa mallin projektista kohdeolion.
void	removeInputPanel (InputPanel inputPanel) Poistaa mallista syöttöpaneelin inputPanel.
void	removeRequirementType (RequirementType requirementType) Poistaa mallin projektista vaatimustyyppin.
void	removeStakeholder (Stakeholder holder) Poistaa mallin projektista sidosryhmän.
void	removeStatus (Status status) Poistaa mallin projektista vaatimuksen tilan.
void	setCurrentUser (User value) Asettaa malliin asiakasohjelman käyttäjän.
void	setDateOfLastUpdate (java.util.Date value) Asettaa viimeisimmän päivityksen ajan.
void	setHistoryEnabled (boolean value) Asettaa historian käytön päälle(true) tai pois päältä (false).
void	setProject (Project project) Asettaa mallin projektiksi projectin.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Model

```
public Model()
```

Konstruktori.

Model

```
public Model(Project project)
```

Konstruktori. Asettaa Mallin Projektiksi projectin.

Method Detail

addRequirement

```
public void addRequirement(Requirement req)
```

Lisää mallin projektiin uuden vaatimuksen tai vanhan vaatimuksen uuden revision riippuen siitä onko mallissa jo vaatimus, jolla on sama id kuin req:lla. Vaatimuksen req olioviitteiden attachments, milestone, manager, lastEditor, authorStakeholder, status, parent ja requirementType pitää osoittaa nukkeolioihin, joiden muut kentät ovat nulleja(tai nollia) paitsi id, joka asetetaan oikeaksi. req:n olioviitte descriptionin pitää osoittaa oikeaan olioon. req:n prioritiesin pitää osoittaa oikeisiin Prioriteetti-olihiin, joiden stakeholder viitteet osoittavat nukkeolioihin. req:n connections-olioviitteiden pitää osoittaa oikeisiin Yhteys-oliioihin, joiden object2-oliiviitteet osoittavat nukkeolioihin. previousVersionin arvolla ei ole väliä, malli huolehtii siitä.

Parameters:

req -

getRequirement

```
public Requirement getRequirement(int reqId)
```

Hakee mallin projektista vaatimuksen, jonka id on reqId. Vanhemmat revisiot löytyvät linkitetystä listasta, jonka juuri uusin revisio on. Listalla liikutaan eteenpäin Vaatimuksen metodilla getPreviousVersion(). Jos vaatimusta ei ole mallissa palautetaan null.

Parameters:

reqId -

Returns:

Requirement

getAllRequirements

```
public java.util.ArrayList<Requirement> getAllRequirements()
```

Hakee mallin projektista kaikki vaatimukset. Vanhemmat revisiot löytyvät linkitetystä listoista, joiden juuria ovat uusimmat revisiot. Listalla liikutaan eteenpäin Vaatimuksen metodilla getPreviousVersion().

Returns:

ArrayList

addUseCase

```
public void addUseCase(UseCase useCase)
```

Lisää malliin projektiin uuden käyttötapauksen tai vanhan käyttötapauksen uuden revision riippuen siitä onko mallissa jo käyttötapaus, jolla on sama id kuin useCase:lla. useCase:n attachmentsin,

lastEditorin ja entitiesin pitää osoittaa nukkeolioihin, joiden muut kentät ovat nulleja(tai nollia) paitsi id, joka asetetaan oikeaksi. useCase:n descriptionin pitää osoittaa oikeaan olioon. useCase:n stakeholdersin pitää osoittaa oikeisiin UseCaseToStakeholder-oliin, joiden stakeholder viitteet osoittavat nukkeolioihin. useCase:n connections-oliiviitteiden pitää osoittaa oikeisiin Yhteys-oliioihin, joiden object2-oliiviitteet osoittavat nukkeolioihin.

Parameters:

useCase -

getUseCase

```
public UseCase getUseCase(int caseId)
```

Hakee mallin projektista käyttötapauksen, jonka id on caseId. Vanhemmat revisiot löytyvät linkitetystä listasta, jonka juuri uusin revisio on. Listalla liikutaan eteenpäin Käyttötapauksen metodilla getPreviousVersion(). Jos käyttötapausta ei ole mallissa palautetaan null.

Parameters:

caseId -

Returns:

UseCase

getAllUseCases

```
public java.util.ArrayList<UseCase> getAllUseCases()
```

Hakee mallin projektista kaikki käyttötapaukset. Vanhemmat revisiot löytyvät linkitetyistä listoista, joiden juuria ovat uusimmat revisiot. Listalla liikutaan eteenpäin Käyttötapauksen metodilla getPreviousVersion().

Returns:

ArrayList

addUser

```
public void addUser(User user)
```

Lisää mallin projektiin uuden käyttäjän, tai korvaa userilla vanhan käyttäjän, jolla on sama id kuin userilla.

Parameters:

user -

getUser

```
public User getUser(int userId)
```

Hakee mallin projektista käyttäjän, jonka id on userId. Jos käyttäjää ei löydy, palautetaan null.

getAllUsers

```
public java.util.ArrayList<User> getAllUsers()
```

Hakee mallin projektista kaikki käyttäjät.

Returns:

ArrayList

addAttachment

```
public void addAttachment(Attachment attachment)
```

Lisää liitetiedoston mallin projektiin, tai korvaa attachmentilla vanhan liitteen, jolla on sama id kuin attachmentilla. Liitetiedostoa ei kiinnitetä vielä mihinkään vaatimukseen/käyttötapaukseen.

Parameters:

attachment -

removeAttachment

```
public void removeAttachment(int attachmentId)
```

Poistaa liitetiedoston projektista. Poisto onnistuu vain, jos liitetiedosto ei ole liitettynä mihinkään vaatimukseen tai käyttötapaukseen.

Parameters:

attachmentId -

getAllAttachments

```
public java.util.List<Attachment> getAllAttachments()
```

Hakee mallin projektista kaikki liitetiedostot.

getAttachment

```
public Attachment getAttachment(int attachmentId)
```

Hakee mallin projektista liitteen, jonka id on attachmentId. Jos liitettä ei löydy, palautetaan null.

addRequirementType

```
public void addRequirementType(RequirementType requirementType)
```

Lisää mallin projektiin uuden vaatimustyyppin, tai korvaa requirementTypellä vanhan vaatimustyyppin,

jolla on sama id kuin requirementTypellä.

Parameters:

requirementType -

getRequirementType

```
public RequirementType getRequirementType(int requirementTypeId)
```

Hakee mallin projektista vaatimustyyppin, jonka id on requirementTypeId.

getAllRequirementTypes

```
public java.util.List<RequirementType> getAllRequirementTypes()
```

Hakee mallin projektista kaikki vaatimustyyppit.

removeRequirementType

```
public void removeRequirementType(RequirementType requirementType)
```

Poistaa mallin projektista vaatimustyyppin. Poisto onnistuu vain, jos vaatimustyyppi ei ole käytössä yhdessäkään vaatimuksessa.

addStatus

```
public void addStatus(Status status)
```

Lisää mallin projektiin uuden vaatimuksen tilan, tai korvaa statuksella vanhan tilan, jolla on sama id kuin statuksella.

getStatus

```
public Status getStatus(int statusId)
```

Hakee mallin projektista vaatimuksen tilan, jonka id on statusId.

getAllStatuses

```
public java.util.List<Status> getAllStatuses()
```

Hakee mallin projektista kaikkien vaatimusten tilat.

removeStatus

```
public void removeStatus(Status status)
```

Poistaa mallin projektista vaatimuksen tilan. Poisto onnistuu vain, jos vaatimuksen tila ei ole käytössä yhdessäkään vaatimuksessa.

addStatistics

```
public void addStatistics(Statistics statistics)
```

Lisää mallin projektiin uuden tilaston, tai korvaa statisticsilla vanhan tilaston, jolla on sama id kuin statisticsilla.

getAllStatistics

```
public java.util.List<Statistics> getAllStatistics()
```

Hakee mallin projektista kaikki tilastot.

addMilestone

```
public void addMilestone(Milestone milestone)
```

Lisää mallin projektiin uuden etapin, tai korvaa milestonella vanhan etapin, jolla on sama id kuin milestonella.

getMilestone

```
public Milestone getMilestone(int milestoneId)
```

Hakee mallin projektista etapin, jonka id on milestoneId.

getAllMilestones

```
public java.util.List<Milestone> getAllMilestones()
```

Hakee mallin projektista kaikki etapit.

addStakeholder

```
public void addStakeholder(Stakeholder holder)
```

Lisää mallin projektiin uuden sidosryhmän, tai korvaa holderilla vanhan sidosryhmän, jolla on sama id kuin holderilla.

getStakeholder

```
public Stakeholder getStakeholder(int stakeholderId)
```

Hakee mallin projektista sidosryhmän, jonka id on stakeholderId.

getAllStakeholders

```
public java.util.List<Stakeholder> getAllStakeholders()
```

Hakee mallin projektista kaikki sidosryhmät.

removeStakeholder

```
public void removeStakeholder(Stakeholder holder)
```

Poistaa mallin projektista sidosryhmän. Poisto onnistuu vain, jos sidosryhmällä ei ole yhteyksiä mihinkään.

setProject

```
public void setProject(Project project)
```

Asettaa mallin projektiksi projectin.

getProject

```
public Project getProject()
```

Hakee mallin projektin.

addEntity

```
public void addEntity(Entity entity)
```

Lisää mallin projektiin uuden kohdeolion, tai korvaa entityllä kohdeolion, jonka id on sama kuin entityllä.

getEntity

```
public Entity getEntity(int entityId)
```

Hakee mallin projektista kohdeolion, jonka id on entityId.

getAllEntities

```
public java.util.List<Entity> getAllEntities()
```

Hakee mallin projektista kaikki kohdeoliot.

removeEntity

```
public void removeEntity(Entity entity)
```

Poistaa mallin projektista kohdeolion. Poisto onnistuu vain, jos kohdeolio ei ole yhdenkään käyttötapausten tai vaatimuksen kohde.

addConnectionType

```
public void addConnectionType(ConnectionType connectionType)
```

Lisää mallin projektiin uuden yhteystyyppin, tai korvaa connectionTypeellä vanhan yhteystyyppin, jolla on sama id kuin connectionTypeellä.

getConnectionType

```
public ConnectionType getConnectionType(int connectionTypeId)
```

Hakee mallin projektista yhteystyyppin, jonka id on connectionTypeId.

getAllConnectionTypes

```
public java.util.List<ConnectionType> getAllConnectionTypes()
```

Hakee mallin projektista kaikki yhteystyyppit.

removeConnectionType

```
public void removeConnectionType(ConnectionType connectionType)
```

Poistaa mallin projektista yhteystyyppin. Poisto onnistuu vain, jos yhteystyyppi ei ole yhdenkään yhteyden tyyppinä.

addInputPanel

```
public void addInputPanel(InputPanel inputPanel)
```

Lisää Malliin uuden syöttöpaneelin, tai korvaa inputPanelilla vanhan syöttöpaneelin, jolla on sama id kuin inputPanelilla.

Parameters:

[inputPanel](#) -

getInputPanel

```
public InputPanel getInputPanel(int inputPanelId)
```

Hakee Mallista Syöttöpaneelin, jonka id on inputPanelId.

Parameters:

inputPanelId -

Returns:

InputPanel

getAllInputPanels

```
public java.util.List<InputPanel> getAllInputPanels()
```

Hakee Mallista kaikki Syöttöpaneelit.

Returns:

List

removeInputPanel

```
public void removeInputPanel(InputPanel inputPanel)
```

Poistaa mallista syöttöpaneelin inputPanel. Poisto onnistuu vain, jos syöttöpaneeli ei ole käytössä yhdessäkään vaatimuksessa tai käyttötapauksessa.

Parameters:

inputPanel -

addDatabase

```
public void addDatabase(Database db)
```

Lisää malliin tietokannan tiedot(nimi, osoite...), tai korvaa db:llä vanhan tietokannan tiedot, joilla on sama id kuin db:llä.

getAllDatabases

```
public java.util.List<Database> getAllDatabases()
```

Hakee mallista kaikkien tietokantojen tiedot.

removeDatabase

```
public void removeDatabase(Database db)
```

Poistaa mallista tietokannan tiedot.

setCurrentUser

```
public void setCurrentUser(User value)
```

Asettaa malliin asiakasohjelman käyttäjän.

getCurrentUser

```
public User getCurrentUser()
```

Hakee parhaillaan sisäänkirjautuneen käyttäjän.

getDateOfLastUpdate

```
public java.util.Date getDateOfLastUpdate()
```

Antaa viimeisimmän päivityksen ajan.

setDateOfLastUpdate

```
public void setDateOfLastUpdate(java.util.Date value)
```

Asettaa viimeisimmän päivityksen ajan.

getHistoryEnabled

```
public boolean getHistoryEnabled()
```

Antaa tiedon siitä, onko historia käytettävissä (true) vai ei (false).

setHistoryEnabled

```
public void setHistoryEnabled(boolean value)
```

Asettaa historian käytön päälle(true) tai pois päältä (false).

[Package](#) **[Class](#)** **[Use Tree](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)**

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Priority

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.Priority

```
public class Priority
extends java.lang.Object
```

Prioriteetti (Priority) on Vaatimukselle annettu painoarvo. Prioriteetilla on aina antajana jokin Sidosryhmä (Stakeholder). Prioriteetti saadaan tietokannan taulusta Priority.

Constructor Summary

[Priority](#)()

Luokan parametrinon konstruktori.

[Priority](#)(int reqRevisionBegin, int reqRevisionEnd, int priority, [Requirement](#) requirement, [Stakeholder](#) stakeholder, int revision)

Luokan konstruktori.

Method Summary

int	getPriority () Get the value of priority
int	getReqRevisionBegin () Get the value of reqRevisionBegin
int	getReqRevisionEnd () Get the value of reqRevisionEnd
Requirement	getRequirement () Get the value of requirement
int	getRevision () Get the value of revision
Stakeholder	getStakeholder () Get the value of stakeholder
void	setPriority (int value) Set the value of priority
void	setReqRevisionBegin (int value) Set the value of reqRevisionBegin

void	setReqRevisionEnd (int value) Set the value of reqRevisionEnd
void	setRequirement (Requirement value) Set the value of requirement
void	setRevision (int value) Set the value of revision
void	setStakeholder (Stakeholder value) Set the value of stakeholder

Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

Priority

```
public Priority()
```

Luokan parametriton konstruktori.

Priority

```
public Priority(int reqRevisionBegin,
               int reqRevisionEnd,
               int priority,
               Requirement requirement,
               Stakeholder stakeholder,
               int revision)
```

Luokan konstruktori. Konstruktoren parametrina annetaan luokan muuttujille arvot.

Method Detail

getReqRevisionBegin

```
public int getReqRevisionBegin()
```

Get the value of reqRevisionBegin

Returns:

the value of reqRevisionBegin

setReqRevisionBegin

```
public void setReqRevisionBegin(int value)
```


Set the value of reqRevisionBegin

getReqRevisionEnd

```
public int getReqRevisionEnd()
```

Get the value of reqRevisionEnd

Returns:

the value of reqRevisionEnd

setReqRevisionEnd

```
public void setReqRevisionEnd(int value)
```

Set the value of reqRevisionEnd

getPriority

```
public int getPriority()
```

Get the value of priority

Returns:

the value of priority

setPriority

```
public void setPriority(int value)
```

Set the value of priority

getRequirement

```
public Requirement getRequirement()
```

Get the value of requirement

Returns:

the value of requirement

setRequirement

```
public void setRequirement(Requirement value)
```

Set the value of requirement

getStakeholder

```
public Stakeholder getStakeholder()
```

Get the value of stakeholder

Returns:

the value of stakeholder

setStakeholder

```
public void setStakeholder(Stakeholder value)
```

Set the value of stakeholder

getRevision

```
public int getRevision()
```

Get the value of revision

Returns:

the value of revision

setRevision

```
public void setRevision(int value)
```

Set the value of revision

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Project

```
java.lang.Object
└─ fi.helsinki.cs.vaativa.model.Project
```

```
public class Project
extends java.lang.Object
```

Projekti-luokka (Project) on keskeisin Mallin luokista. Kaikki muut luokat liittyvät siihen. Projekti-oliosta on pääsy - mahdollisesti mutkan kautta - kaikkiin Mallin luokkiin. Yhtä Mallia kohti on aina yksi Projekti. Projektin on yksikkö, jonka kautta Vaativassa jäsennetään vaatimustenhallintaa. Yhden ohjelmistotuotantoprojektin tiedot kuuluvat aina yhteen Projekti-oliioon.

Field Summary

static java.lang.String	CLOSED Projektiin tilan kuvaamisessa käytettävä arvo, joka tarkoittaa että Projekti on suljettu, eli on mahdollista ainoastaan katsella Projektin tietoja.
static java.lang.String	OPEN Projektiin tilan kuvaamisessa käytettävä arvo, joka tarkoittaa että Projekti on avoin, eli siihen voidaan mm. lisätä vaatimuksia.

Constructor Summary

[Project](#)()

Luokan parametrin konstruktori.

```
Project(java.util.ArrayList<Attachment> attachments, java.util.Date closed,
java.util.ArrayList<ConnectionType> connectionTypes, java.util.Date created,
java.lang.String description, java.util.ArrayList<Entity> entities, int id,
java.util.ArrayList<InputPanel> inputPanels, int maxPriority,
java.util.ArrayList<Milestone> milestones, java.lang.String name, java.lang.String password,
java.util.ArrayList<Requirement> requirements,
java.util.ArrayList<RequirementType> requirementTypes,
java.util.ArrayList<Stakeholder> stakeholders, java.util.ArrayList<Statistics> statistics,
java.lang.String status, java.util.ArrayList<Status> statuses,
java.util.ArrayList<UseCase> usecases, java.util.ArrayList<User> users)
```

Luokan konstruktori.

Method Summary

java.util.ArrayList< Attachment >	getAllAttachments () Get Attachments.
java.util.ArrayList< ConnectionType >	getAllConnectionTypes () Get ConnectionTypes.
java.util.ArrayList< Entity >	getAllEntities () Get Entities.

java.util.ArrayList< InputPanel >	getAllInputPanels () Get InputPanels.
java.util.ArrayList< Milestone >	getAllMilestones () Get Milestones.
java.util.ArrayList< Requirement >	getAllRequirements () Get requirements
java.util.ArrayList< RequirementType >	getAllRequirementTypes () Get RequirementTypes.
java.util.ArrayList< Stakeholder >	getAllStakeholders () Get Stakeholders.
java.util.ArrayList< Statistics >	getAllStatistics () Get Statistics.
java.util.ArrayList< Status >	getAllStatuses () Get Statuses.
java.util.ArrayList< UseCase >	getAllUseCases () Get useCases.
java.util.ArrayList< User >	getAllUsers () Get Users.
java.util.Date	getClosed () Get the value of closed
java.util.Date	getCreated () Get the value of created
java.lang.String	getDescription () Get the value of description
int	getId () Get the value of id
int	getMaxPriority () Get the value of maxPriority
java.lang.String	getName () Get the value of name
java.lang.String	getPassword () Get the value of password
java.lang.String	getStatus () Get the value of status
void	setClosed (java.util.Date value) Set the value of closed
void	setConnectionTypes (java.util.ArrayList< ConnectionType > value) Set the value of ConnectionTypes.
void	setCreated (java.util.Date value) Set the value of created
void	setDescription (java.lang.String value) Set the value of description
void	setEntities (java.util.ArrayList< Entity > value) Set the value of Entities.

void	<u>setId</u> (int value) Set the value of id
void	<u>setInputPanels</u> (java.util.ArrayList< <u>InputPanel</u> > value) Set the value of InputPanels.
void	<u>setMaxPriority</u> (int value) Set the value of maxPriority
void	<u>setMilestones</u> (java.util.ArrayList< <u>Milestone</u> > value) Set the value of Milestones.
void	<u>setName</u> (java.lang.String value) Set the value of name
void	<u>setPassword</u> (java.lang.String value) Set the value of password
void	<u>setRequirements</u> (java.util.ArrayList< <u>Requirement</u> > value) Set the value of requirements.
void	<u>setRequirementTypes</u> (java.util.ArrayList< <u>RequirementType</u> > value) Set the value of requirementTypes.
void	<u>setStakeholders</u> (java.util.ArrayList< <u>Stakeholder</u> > value) Set the value of Stakeholders.
void	<u>setStatus</u> (java.lang.String value) Set the value of status
void	<u>setUseCases</u> (java.util.ArrayList< <u>UseCase</u> > value) Set the value of useCases.
void	<u>setUsers</u> (java.util.ArrayList< <u>User</u> > value) Set the value of Users.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

OPEN

```
public static final java.lang.String OPEN
```

Projektin tilan kuvaamisessa käytettävä arvo, joka tarkoittaa että Projekti on avoin, eli siihen voidaan mm. lisätä vaatimuksia.

See Also:

[Constant Field Values](#)

CLOSED

```
public static final java.lang.String CLOSED
```

Projektin tilan kuvaamisessa käytettävä arvo, joka tarkoittaa että Projekti on suljettu, eli on mahdollista ainoastaan katsella Projektin tietoja.

See Also:

[Constant Field Values](#)

Constructor Detail

Project

```
public Project()
```

Luokan parametrin konstruktori.

Project

```
public Project(java.util.ArrayList<Attachment> attachments,  
              java.util.Date closed,  
              java.util.ArrayList<ConnectionType> connectionTypes,  
              java.util.Date created,  
              java.lang.String description,  
              java.util.ArrayList<Entity> entities,  
              int id,  
              java.util.ArrayList<InputPanel> inputPanels,  
              int maxPriority,  
              java.util.ArrayList<Milestone> milestones,  
              java.lang.String name,  
              java.lang.String password,  
              java.util.ArrayList<Requirement> requirements,  
              java.util.ArrayList<RequirementType> requirementTypes,  
              java.util.ArrayList<Stakeholder> stakeholders,  
              java.util.ArrayList<Statistics> statistics,  
              java.lang.String status,  
              java.util.ArrayList<Status> statuses,  
              java.util.ArrayList<UseCase> usecases,  
              java.util.ArrayList<User> users)
```

Luokan konstruktori. Konstruktorin parametrina annetaan luokan muuttujille arvot.

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getName

```
public java.lang.String getName()
```

Get the value of name

Returns:

the value of name

getAllStatuses

```
public java.util.ArrayList<Status> getAllStatuses()
```

Get Statuses.

Returns:

all Statuses

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getDescription

```
public java.lang.String getDescription()
```

Get the value of description

Returns:

the value of description

setDescription

```
public void setDescription(java.lang.String value)
```

Set the value of description

getPassword

```
public java.lang.String getPassword()
```

Get the value of password

Returns:

the value of password

setPassword

```
public void setPassword(java.lang.String value)
```

Set the value of password

getMaxPriority

```
public int getMaxPriority()
```

Get the value of maxPriority

Returns:

the value of maxPriority

setMaxPriority

```
public void setMaxPriority(int value)
```

Set the value of maxPriority

getStatus

```
public java.lang.String getStatus()
```

Get the value of status

Returns:

the value of status

setStatus

```
public void setStatus(java.lang.String value)
```

Set the value of status

getCreated

```
public java.util.Date getCreated()
```

Get the value of created

Returns:

the value of created

setCreated

```
public void setCreated(java.util.Date value)
```

Set the value of created

getClosed

```
public java.util.Date getClosed()
```

Get the value of closed

Returns:

the value of closed

setClosed

```
public void setClosed(java.util.Date value)
```

Set the value of closed

getAllRequirements

```
public java.util.ArrayList<Requirement> getAllRequirements()
```

Get requirements

Returns:

all requirements

setRequirements

```
public void setRequirements(java.util.ArrayList<Requirement> value)
```

Set the value of requirements.

getAllStatistics

```
public java.util.ArrayList<Statistics> getAllStatistics()
```

Get Statistics.

Returns:

all Statistics

getAllUsers

```
public java.util.ArrayList<User> getAllUsers()
```

Get Users.

Returns:

all Userrs

setUsers

```
public void setUsers(java.util.ArrayList<User> value)
```

Set the value of Users.

getAllUseCases

```
public java.util.ArrayList<UseCase> getAllUseCases()
```

Get useCases.

Returns:

all UseCases

setUseCases

```
public void setUseCases(java.util.ArrayList<UseCase> value)
```

Set the value of useCases.

getAllMilestones

```
public java.util.ArrayList<Milestone> getAllMilestones()
```

Get Milestones.

Returns:

all Milestones

setMilestones

```
public void setMilestones(java.util.ArrayList<Milestone> value)
```

Set the value of Milestones.

getAllStakeholders

```
public java.util.ArrayList<Stakeholder> getAllStakeholders()
```

Get Stakeholders.

Returns:

all Stakeholders

setStakeholders

```
public void setStakeholders(java.util.ArrayList<Stakeholder> value)
```

Set the value of Stakeholders.

getAllEntities

```
public java.util.ArrayList<Entity> getAllEntities()
```

Get Entities.

Returns:

all Entities

setEntities

```
public void setEntities(java.util.ArrayList<Entity> value)
```

Set the value of Entities.

getAllConnectionTypes

```
public java.util.ArrayList<ConnectionType> getAllConnectionTypes()
```

Get ConnectionTypes.

Returns:

all ConnectionTypes

setConnectionTypes

```
public void setConnectionTypes(java.util.ArrayList<ConnectionType> value)
```

Set the value of ConnectionTypes.

getAllInputPanels

```
public java.util.ArrayList<InputPanel> getAllInputPanels()
```

Get InputPanels.

Returns:

all InputPanels.

setInputPanels

```
public void setInputPanels(java.util.ArrayList<InputPanel> value)
```

Set the value of InputPanels.

getAllRequirementTypes

```
public java.util.ArrayList<RequirementType> getAllRequirementTypes()
```

Get RequirementTypes.

Returns:

all RequirementTypes.

setRequirementTypes

```
public void setRequirementTypes(java.util.ArrayList<RequirementType> value)
```

Set the value of requirementTypes.

getAllAttachments

```
public java.util.ArrayList<Attachment> getAllAttachments()
```

Get Attachments.

Returns:

all Attachments.

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Requirement

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.Requirement

All Implemented Interfaces:

[Connectable](#)

```
public class Requirement
extends java.lang.Object
implements Connectable
```

Vaatus on keskeinen Mallin luokka. Se kapseloi sisäänsä järjestelmään syötetyn vaatimuksen tiedot. Vaatus luodaan tietokannan Requirement-taulun perusteella.

Constructor Summary

[Requirement](#)()

Luokan parametrin konstruktori.

[Requirement](#)(java.util.ArrayList<[Attachment](#)> attachments, java.lang.String authorName, [Stakeholder](#) authorStakeholder, java.util.ArrayList<[Connection](#)> connections, java.util.Date created, [Description](#) description, int id, [User](#) lastEditor, [User](#) manager, [Milestone](#) milestone, java.util.Date modified, java.lang.String name, java.lang.String notes, [Requirement](#) parent, [Requirement](#) previousVersion, java.util.ArrayList<[Priority](#)> priorities, int revision, [Status](#) status, [RequirementType](#) type)

Luokan konstruktori.

Method Summary

java.util.ArrayList< Attachment >	getAllAttachments () Get the value of attachments
java.util.ArrayList< Connection >	getAllConnections () Get the value of connections
java.util.ArrayList< Entity >	getAllEntities () Get the value of entity
java.util.ArrayList< Priority >	getAllPriorities () Get the value of priorities
java.lang.String	getAuthorName () Get the value of authorName

Stakeholder	getAuthorStakeholder () Get the value of authorStakeholder
java.util.Date	getCreated () Get the value of created
Description	getDescription () Get the value of description
int	getId () Get the value of id
User	getLastEditor () Get the value of lastEditor
User	getManager () Get the value of manager
Milestone	getMilestone () Get the value of milestone
java.util.Date	getModified () Get the value of modified
java.lang.String	getName () Get the value of name
java.lang.String	getNotes () Get the value of notes
Requirement	getParent () Get the value of parent
Requirement	getPreviousVersion () Get the value of previousVersion
int	getRevision () Get the value of revision
Status	getStatus () Get the value of status
RequirementType	getType () Get the value of type
void	setAttachments (java.util.ArrayList< Attachment > value) Set the value of attachments
void	setAuthorName (java.lang.String value) Set the value of authorName
void	setAuthorStakeholder (Stakeholder value) entities = value; Set the value of authorStakeholder
void	setConnections (java.util.ArrayList< Connection > value) Set the value of connections
void	setCreated (java.util.Date value) Set the value of created

void	<u>setDescription</u> (<u>Description</u> value) Set the value of description
void	<u>setEntities</u> (java.util.ArrayList< <u>Entity</u> > value) Set the value of entity
void	<u>setId</u> (int value) Set the value of id
void	<u>setLastEditor</u> (<u>User</u> value) Set the value of lastEditor
void	<u>setManager</u> (<u>User</u> value) Set the value of manager
void	<u>setMilestone</u> (<u>Milestone</u> value) Set the value of milestone
void	<u>setModified</u> (java.util.Date value) Set the value of modified
void	<u>setName</u> (java.lang.String value) Set the value of name
void	<u>setNotes</u> (java.lang.String value) Set the value of notes
void	<u>setParent</u> (<u>Requirement</u> value) Set the value of parent
void	<u>setPreviousVersion</u> (<u>Requirement</u> value) Set the value of previousVersion
void	<u>setPriorities</u> (java.util.ArrayList< <u>Priority</u> > value) Set the value of priorities
void	<u>setRevision</u> (int value) Set the value of revision
void	<u>setStatus</u> (<u>Status</u> value) Set the value of status
void	<u>setType</u> (<u>RequirementType</u> value) Set the value of type

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Requirement

```
public Requirement()
```

Luokan parametrin konstruktori.

Requirement

```
public Requirement(java.util.ArrayList<Attachment> attachments,
    java.lang.String authorName,
    Stakeholder authorStakeholder,
    java.util.ArrayList<Connection> connections,
    java.util.Date created,
    Description description,
    int id,
    User lastEditor,
    User manager,
    Milestone milestone,
    java.util.Date modified,
    java.lang.String name,
    java.lang.String notes,
    Requirement parent,
    Requirement previousVersion,
    java.util.ArrayList<Priority> priorities,
    int revision,
    Status status,
    RequirementType type)
```

Luokan konstruktori. Konstruktoria parametrina annetaan luokan kaikille muuttujille arvot.

Method Detail

getId

```
public int getId()
```

Get the value of id

Specified by:

[getId](#) in interface [Connectable](#)

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getName

```
public java.lang.String getName()
```

Get the value of name

Specified by:

[getName](#) in interface [Connectable](#)

Returns:

the value of name

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getStatus

```
public Status getStatus()
```

Get the value of status

Returns:

the value of status

setStatus

```
public void setStatus(Status value)
```

Set the value of status

getNotes

```
public java.lang.String getNotes()
```

Get the value of notes

Returns:

the value of notes

setNotes

```
public void setNotes(java.lang.String value)
```

Set the value of notes

getCreated

```
public java.util.Date getCreated()
```

Get the value of created

Returns:

the value of created

setCreated

```
public void setCreated(java.util.Date value)
```

Set the value of created

getModified

```
public java.util.Date getModified()
```

Get the value of modified

Returns:

the value of modified

setModified

```
public void setModified(java.util.Date value)
```

Set the value of modified

getType

```
public RequirementType getType()
```

Get the value of type

Returns:

the value of type

setType

```
public void setType(RequirementType value)
```

Set the value of type

getMilestone

```
public Milestone getMilestone()
```

Get the value of milestone

Returns:

the value of milestone

setMilestone

```
public void setMilestone(Milestone value)
```

Set the value of milestone

getManager

```
public User getManager()
```

Get the value of manager

Returns:

the value of manager

setManager

```
public void setManager(User value)
```

Set the value of manager

getAuthorName

```
public java.lang.String getAuthorName()
```

Get the value of authorName

Returns:

the value of authorName

setAuthorName

```
public void setAuthorName(java.lang.String value)
```

Set the value of authorName

getAuthorStakeholder

```
public Stakeholder getAuthorStakeholder()
```

Get the value of authorStakeholder

Returns:

the value of authorStakeholder

setAuthorStakeholder

```
public void setAuthorStakeholder(Stakeholder value)
```

entities = value; Set the value of authorStakeholder

getLastEditor

```
public User getLastEditor()
```

Get the value of lastEditor

Returns:

the value of lastEditor

setLastEditor

```
public void setLastEditor(User value)
```

Set the value of lastEditor

getRevision

```
public int getRevision()
```

Get the value of revision

Returns:

the value of revision

setRevision

```
public void setRevision(int value)
```

Set the value of revision

getAllAttachments

```
public java.util.ArrayList<Attachment> getAllAttachments()
```

Get the value of attachments

Returns:

the value of attachments

setAttachments

```
public void setAttachments(java.util.ArrayList<Attachment> value)
```

Set the value of attachments

getParent

```
public Requirement getParent()
```

Get the value of parent

Returns:

the value of parent

setParent

```
public void setParent(Requirement value)
```

Set the value of parent

getDescription

```
public Description getDescription()
```

Get the value of description

Returns:

the value of description

setDescription

```
public void setDescription(Description value)
```

Set the value of description

setEntities

```
public void setEntities(java.util.ArrayList<Entity> value)
```

Set the value of entity

getAllEntities

```
public java.util.ArrayList<Entity> getAllEntities()
```

Get the value of entity

Returns:

the value of entities

setPreviousVersion

```
public void setPreviousVersion(Requirement value)
```

Set the value of previousVersion

getPreviousVersion

```
public Requirement getPreviousVersion()
```

Get the value of previousVersion

Returns:

the value of previousVersion

setConnections

```
public void setConnections(java.util.ArrayList<Connection> value)
```

Set the value of connections

getAllConnections

```
public java.util.ArrayList<Connection> getAllConnections()
```

Get the value of connections

Returns:

the value of connections

setPriorities

```
public void setPriorities(java.util.ArrayList<Priority> value)
```

Set the value of priorities

getAllPriorities

```
public java.util.ArrayList<Priority> getAllPriorities()
```

Get the value of priorities

Returns:

the value of priorities

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class RequirementType

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.RequirementType

```
public class RequirementType
extends java.lang.Object
```

VaatusTyyppi (RequirementType) on tyyppi, mikä voidaan antaa Vaatimukselle. Jokaisella Vaatimuksella on yksi VaatimusTyyppi. VaatimusTyyppit voivat muodostaa hierarkkisen puurakenteen, jossa esimerkiksi käyttäjävaatimus-VaatimusTyyppi toimii järjestelmävaatimus-VaatimusTyyppin vanhempana. Vaatimustyyppiin liittyy oletussyöttöpaneeli, joka haetaan liitännäisjärjestelmästä name:n perusteella.

Constructor Summary

[RequirementType](#)()

Luokan parametriton konstruktori.

[RequirementType](#)(int id, java.lang.String name, java.lang.String description, java.lang.String defaultEntryPanelName, [RequirementType](#) parent)

Luokan konstruktori.

Method Summary

java.lang.String	getDefaultEntryPanelName () Get the value of defaultEntryPanelId
java.lang.String	getDescription () Get the value of description
int	getId () Get the value of id
java.lang.String	getName () Get the value of name
RequirementType	getParent () Get the value of parent
void	setDefaultEntryPanelName (java.lang.String value) Set the value of defaultEntryPanelId
void	setDescription (java.lang.String value) Set the value of description
void	setId (int value) Set the value of id

void	setName (java.lang.String value) Set the value of name
void	setParent (RequirementType value) Set the value of parent

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

RequirementType

```
public RequirementType()
```

Luokan parametrin konstruktori.

RequirementType

```
public RequirementType(int id,  
    java.lang.String name,  
    java.lang.String description,  
    java.lang.String defaultEntryPanelName,  
    RequirementType parent)
```

Luokan konstruktori. Konstruktorin parametrina annetaan luokan muuttujille arvot.

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getName


```
public java.lang.String getName()
```

Get the value of name

Returns:

the value of name

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getDescription

```
public java.lang.String getDescription()
```

Get the value of description

Returns:

the value of description

setDescription

```
public void setDescription(java.lang.String value)
```

Set the value of description

getDefaultEntryPanelName

```
public java.lang.String getDefaultEntryPanelName()
```

Get the value of defaultEntryPanelId

Returns:

the value of defaultEntryPanelId

setDefaultEntryPanelName

```
public void setDefaultEntryPanelName(java.lang.String value)
```

Set the value of defaultEntryPanelId

getParent

```
public RequirementType getParent()
```

Get the value of parent

Returns:

the value of parent

setParent

```
public void setParent(RequirementType value)
```

Set the value of parent

[Package](#) **[Class](#)** **[Use Tree](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)**

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Stakeholder

java.lang.Object
└─ fi.helsinki.cs.vaativa.model.Stakeholder

```
public class Stakeholder  
extends java.lang.Object
```

Sidosryhmä (Stakeholder) on järjestelmään liittyvän sidosryhmän tietojen kapselointi luokkaan. Sidosryhmät antavat Vaatimuksille Prioriteetteja, liittyvät Käyttötapauksiin ja ovat mukana Projektissa jossain roolissa.

Constructor Summary

[Stakeholder](#)()
Konstruktori

[Stakeholder](#)(int id, java.lang.String name, java.lang.String description,
java.util.ArrayList<[UseCaseToStakeholder](#)> useCases)
Konstruktori.

Method Summary

java.util.ArrayList< UseCaseToStakeholder >	getAllUseCases () Get the value of usecases
java.lang.String	getDescription () Get the value of description
int	getId () Get the value of id
java.lang.String	getName () Get the value of name
void	setDescription (java.lang.String value) Set the value of description
void	setId (int value) Set the value of id
void	setName (java.lang.String value) Set the value of name
void	setUseCases (java.util.ArrayList< UseCaseToStakeholder > value) Set the value of usecases

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Stakeholder

```
public Stakeholder(int id,
                   java.lang.String name,
                   java.lang.String description,
                   java.util.ArrayList<UseCaseToStakeholder> useCases)
```

Konstruktori.

Stakeholder

```
public Stakeholder()
```

Konstruktori

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getName

```
public java.lang.String getName()
```

Get the value of name

Returns:

the value of name

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getDescription

```
public java.lang.String getDescription()
```

Get the value of description

Returns:

the value of description

setDescription

```
public void setDescription(java.lang.String value)
```

Set the value of description

getAllUseCases

```
public java.util.ArrayList<UseCaseToStakeholder> getAllUseCases()
```

Get the value of usecases

Returns:

the value of usecases

setUseCases

```
public void setUseCases(java.util.ArrayList<UseCaseToStakeholder> value)
```

Set the value of usecases

[Package](#) **[Class](#)** **[Use](#)** **[Tree](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)**

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Statistics

java.lang.Object

└ [fi.helsinki.cs.vaativa.model.Statistics](#)

```
public class Statistics  
extends java.lang.Object
```

Tilastot-luokka kapseloi sisäänsä tiedot siitä, motako tietyn tyyppistä Vaatimusta oli tiettyssä Tilassa tiettyinä päivänä.

Constructor Summary

[Statistics](#)()

Luokan parametriton konstruktori.

[Statistics](#)(int id, java.util.Date date, int amount, [Requirement](#) type, [Status](#) status)

Luokan konstruktori.

Method Summary

int	getAmount () Get the value of amount
java.util.Date	getDate () Get the value of date
int	getId () Get the value of id
Status	getStatus () Get the value of status
Requirement	getType () Get the value of type
void	setAmount (int value) Set the value of amount
void	setDate (java.util.Date value) Set the value of date
void	setId (int value) Set the value of id
void	setStatus (Status value) Set the value of status

void	setType (Requirement value) Set the value of type
------	---

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Statistics

```
public Statistics()
```

Luokan parametriton konstruktori.

Statistics

```
public Statistics(int id,  
                 java.util.Date date,  
                 int amount,  
                 Requirement type,  
                 Status status)
```

Luokan konstruktori. Konstruktoren parametrina annetaan luokan muuttujille arvot.

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getDate

```
public java.util.Date getDate()
```

Get the value of date

Returns:

the value of date

setDate

```
public void setDate(java.util.Date value)
```

Set the value of date

getAmount

```
public int getAmount()
```

Get the value of amount

Returns:

the value of amount

setAmount

```
public void setAmount(int value)
```

Set the value of amount

getType

```
public Requirement getType()
```

Get the value of type

Returns:

the value of type

setType

```
public void setType(Requirement value)
```

Set the value of type

getStatus

```
public Status getStatus()
```

Get the value of status

Returns:

the value of status

setStatus

```
public void setStatus(Status value)
```

Set the value of status

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class Status

java.lang.Object

└─ **fi.helsinki.cs.vaativa.model.Status**

```
public class Status
extends java.lang.Object
```

Tila (Status) on tila, jossa Vaatimus on. Tila voisi olla esimerkiksi hylätty, odottaa vahvistusta tai valmis. Projektissa pidetään listaa Tiloista, jotta käyttäjille voidaan antaa järkeviä oletusarvoja ja Vaatimusten tilojen hallinta on yhtenäistä.

Constructor Summary

[Status](#)()

Luokan parametriton konstruktori.

[Status](#)(int id, java.lang.String name, java.lang.String description)

Luokan konstruktori.

Method Summary

java.lang.String	getDescription () Get the value of description
int	getId () Get the value of id
java.lang.String	getName () Get the value of name
void	setDescription (java.lang.String value) Set the value of description
void	setId (int value) Set the value of id
void	setName (java.lang.String value) Set the value of name

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

Status

```
public Status()
```

Luokan parametrin konstruktori.

Status

```
public Status(int id,  
              java.lang.String name,  
              java.lang.String description)
```

Luokan konstruktori. Konstruktorin parametrina annetaan luokan muuttujille arvot.

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getName

```
public java.lang.String getName()
```

Get the value of name

Returns:

the value of name

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getDescription

```
public java.lang.String getDescription()
```

Get the value of description

Returns:

the value of description

setDescription

```
public void setDescription(java.lang.String value)
```

Set the value of description

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class UseCase

```
java.lang.Object  
└─ fi.helsinki.cs.vaativa.model.UseCase
```

All Implemented Interfaces:

[Connectable](#)

```
public class UseCase  
extends java.lang.Object  
implements Connectable
```

Käyttötapaus (UseCase) on järjestelmään kirjatun käyttötapausten kapselointi luokaksi. Käyttötapaus on yksi Mallin keskeisimmistä luokista.

Constructor Summary

[UseCase](#) ()

Konstruktori

[UseCase](#) (int id, java.util.ArrayList<[Attachment](#)> attachments, java.util.ArrayList<[Connection](#)> connections, java.util.Date created, [Description](#) description, [User](#) lastEditor, java.util.Date modified, java.lang.String name, java.lang.String notes, [UseCase](#) previousVersion, int revision, java.util.ArrayList<[UseCaseToStakeholder](#)> stakeholders)
Konstruktori.

Method Summary

java.util.ArrayList< Attachment >	getAllAttachments () Get the value of attachments
java.util.ArrayList< Connection >	getAllConnections () Get the value of connections
java.util.ArrayList< Entity >	getAllEntities () Get the value of entity
java.util.ArrayList< UseCaseToStakeholder >	getAllStakeholders () Get the value of stakeholders
java.util.Date	getCreated () Get the value of created
Description	getDescription () Get the value of description
int	getId () Get the value of id
User	getLastEditor () Get the value of lastEditor
java.util.Date	getModified () Get the value of modified

java.lang.String	getName() Get the value of name
java.lang.String	getNotes() Get the value of notes
UseCase	getPreviousVersion() Get the value of previousVersion
int	getRevision() Get the value of revision
void	setAttachments() (java.util.ArrayList< Attachment > value) Set the value of attachments
void	setConnections() (java.util.ArrayList< Connection > value) Set the value of connections
void	setCreated() (java.util.Date value) Set the value of created
void	setDescription() (Description value) Set the value of description
void	setEntities() (java.util.ArrayList< Entity > value) Set the value of entity
void	setId() (int value) Set the value of id
void	setLastEditor() (User value) Set the value of lastEditor
void	setModified() (java.util.Date value) Set the value of modified
void	setName() (java.lang.String value) Set the value of name
void	setNotes() (java.lang.String value) Set the value of notes
void	setPreviousVersion() (UseCase value) Set the value of previousVersion
void	setRevision() (int value) Set the value of revision
void	setStakeholders() (java.util.ArrayList< UseCaseToStakeholder > value) Set the value of stakeholders

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

UseCase

```
public UseCase(int id,
    java.util.ArrayList<Attachment> attachments,
    java.util.ArrayList<Connection> connections,
    java.util.Date created,
    Description description,
    User lastEditor,
    java.util.Date modified,
```

```
java.lang.String name,  
java.lang.String notes,  
UseCase previousVersion,  
int revision,  
java.util.ArrayList<UseCaseToStakeholder> stakeholders)
```

Konstruktori.

UseCase

```
public UseCase()
```

Konstruktori

Method Detail

getId

```
public int getId()
```

Get the value of id

Specified by:

[getId](#) in interface [Connectable](#)

Returns:

the value of id

setId

```
public void setId(int value)
```

Set the value of id

getName

```
public java.lang.String getName()
```

Get the value of name

Specified by:

[getName](#) in interface [Connectable](#)

Returns:

the value of name

setName

```
public void setName(java.lang.String value)
```

Set the value of name

getCreated

```
public java.util.Date getCreated()
```

Get the value of created

Returns:

the value of created

setCreated

```
public void setCreated(java.util.Date value)
```

Set the value of created

getModified

```
public java.util.Date getModified()
```

Get the value of modified

Returns:

the value of modified

setModified

```
public void setModified(java.util.Date value)
```

Set the value of modified

getNotes

```
public java.lang.String getNotes()
```

Get the value of notes

Returns:

the value of notes

setNotes

```
public void setNotes(java.lang.String value)
```

Set the value of notes

getRevision

```
public int getRevision()
```

Get the value of revision

Returns:

the value of revision

setRevision

```
public void setRevision(int value)
```

Set the value of revision

getAllStakeholders

```
public java.util.ArrayList<UseCaseToStakeholder> getAllStakeholders()
```

Get the value of stakeholders

Returns:

the value of stakeholders

setStakeholders

```
public void setStakeholders(java.util.ArrayList<UseCaseToStakeholder> value)
```

Set the value of stakeholders

getLastEditor

```
public User getLastEditor()
```

Get the value of lastEditor

Returns:

the value of lastEditor

setLastEditor

```
public void setLastEditor(User value)
```

Set the value of lastEditor

getAllAttachments

```
public java.util.ArrayList<Attachment> getAllAttachments()
```

Get the value of attachments

Returns:

the value of attachments

setAttachments

```
public void setAttachments(java.util.ArrayList<Attachment> value)
```

Set the value of attachments

getDescription

```
public Description getDescription()
```

Get the value of description

Returns:

the value of description

setDescription

```
public void setDescription(Description value)
```

Set the value of description

setPreviousVersion

```
public void setPreviousVersion(UseCase value)
```

Set the value of previousVersion

getPreviousVersion

```
public UseCase getPreviousVersion()
```

Get the value of previousVersion

Returns:

the value of previousVersion

setEntities

```
public void setEntities(java.util.ArrayList<Entity> value)
```

Set the value of entity

getAllEntities

```
public java.util.ArrayList<Entity> getAllEntities()
```

Get the value of entity

Returns:

the value of entities

setConnections

```
public void setConnections(java.util.ArrayList<Connection> value)
```

Set the value of connections

getAllConnections

```
public java.util.ArrayList<Connection> getAllConnections()
```

Get the value of connections

Returns:

the value of connections

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class UseCaseToStakeholder

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.UseCaseToStakeholder

```
public class UseCaseToStakeholder
extends java.lang.Object
```

UseCaseToStakeholder kuvaa Sidosryhmien ja Käyttötapausten yhteyttä.

Constructor Summary

[UseCaseToStakeholder](#)()

Konstruktori

[UseCaseToStakeholder](#)([UseCase](#) useCase, [Stakeholder](#) stakeholder, java.lang.String role)

Konstruktori.

Method Summary

java.lang.String	getRole () Get the value of role
Stakeholder	getStakeholder () Get the value of stakeholder
UseCase	getUseCase () Get the value of useCase
void	setRole (java.lang.String value) Set the value of role
void	setStakeholder (Stakeholder value) Set the value of stakeholder
void	setUseCase (UseCase value) Set the value of useCase

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

UseCaseToStakeholder

```
public UseCaseToStakeholder(UseCase useCase,  
                             Stakeholder stakeholder,  
                             java.lang.String role)
```

Konstruktor.

UseCaseToStakeholder

```
public UseCaseToStakeholder()
```

Konstruktor

Method Detail

getUseCase

```
public UseCase getUseCase()
```

Get the value of useCase

Returns:

the value of useCase

setUseCase

```
public void setUseCase(UseCase value)
```

Set the value of useCase

getStakeholder

```
public Stakeholder getStakeholder()
```

Get the value of stakeholder

Returns:

the value of stakeholder

setStakeholder

```
public void setStakeholder(Stakeholder value)
```

Set the value of stakeholder

getRole

```
public java.lang.String getRole()
```

Get the value of role

Returns:

the value of role

setRole

```
public void setRole(java.lang.String value)
```

Set the value of role

[Package](#) **[Class](#)** **[Use Tree](#)** **[Deprecated](#)** **[Index](#)** **[Help](#)**

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Class User

java.lang.Object

└─ fi.helsinki.cs.vaativa.model.User

```
public class User
extends java.lang.Object
```

Käyttäjä (User) on järjestelmään kirjautuneen käyttäjän tietojen kapselointi luokaksi. Käyttäjä saadaan luotua tietokannan User-taulun avulla.

Nested Class Summary

static class	User.UserStatus Kelvolliset arvot Käyttäjän tilalle: ACTIVE= Käyttäjän tunnus on käytössä INACTIVE= ylläpitäjä on jäädyttänyt tunnuksen AWAITING_AUTHORIZATION= tunnus odottaa ylläpitäjän hyväksyntää
--------------	--

Constructor Summary

[User](#)()
Luokan parametriton konstruktori.

[User](#)(java.util.Date created, java.lang.String email, java.util.Date exitDate, java.util.Date expire, java.lang.String firstName, int id, java.util.Date joinDate, java.lang.String lastName, java.lang.String password, java.lang.String role, java.lang.String settingsXML, [Stakeholder](#) stakeholder, [Status](#) status, int type, java.lang.String userName)
Luokan konstruktori.

Method Summary

java.util.Date	getCreated () Get the value of created
java.lang.String	getEmail () Get the value of email
java.util.Date	getExitDate () Get the value of exitDate
java.util.Date	getExpire () Get the value of expire

java.lang.String	<u>getFirstName()</u> Get the value of firstName
int	<u>getId()</u> Get the value of id
java.util.Date	<u>getJoinDate()</u> Get the value of joinDate
java.lang.String	<u>getLastName()</u> Get the value of lastName
java.lang.String	<u>getPassword()</u> Get the value of password
java.lang.String	<u>getRole()</u> Get the value of role
java.lang.String	<u>getSettingsXML()</u> Get the value of settingsXML
Status	<u>getStatus()</u> Get the value of status
int	<u>getType()</u> Get the value of type
java.lang.String	<u>getUserName()</u> Get the value of userName
void	<u>setCreated()</u> (java.util.Date value) Set the value of created
void	<u>setEmail()</u> (java.lang.String value) Set the value of email
void	<u>setExitDate()</u> (java.util.Date value) Set the value of exitDate
void	<u>setExpire()</u> (java.util.Date value) Set the value of expire
void	<u>setFirstName()</u> (java.lang.String value) Set the value of firstName
void	<u>setId()</u> (int value) Set the value of id
void	<u>setJoinDate()</u> (java.util.Date value) Set the value of joinDate
void	<u>setLastName()</u> (java.lang.String value) Set the value of lastName
void	<u>setPassword()</u> (java.lang.String value) Set the value of password
void	<u>setRole()</u> (java.lang.String value) Set the value of role

void	setSettingsXML (java.lang.String value) Set the value of settingsXML
void	setStatus (Status value) Set the value of status
void	setType (int value) Set the value of type
void	setUserName (java.lang.String value) Set the value of userName

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

User

```
public User()
```

Luokan parametrin konstruktori.

User

```
public User(java.util.Date created,
            java.lang.String email,
            java.util.Date exitDate,
            java.util.Date expire,
            java.lang.String firstName,
            int id,
            java.util.Date joinDate,
            java.lang.String lastName,
            java.lang.String password,
            java.lang.String role,
            java.lang.String settingsXML,
            Stakeholder stakeholder,
            Status status,
            int type,
            java.lang.String userName)
```

Luokan konstruktori. Konstruktorin parametrina annetaan luokan muuttujille arvot.

Method Detail

getId

```
public int getId()
```

Get the value of id

Returns:
the value of id

setId

```
public void setId(int value)
```

Set the value of id

getFirstName

```
public java.lang.String getFirstName()
```

Get the value of firstName

Returns:
the value of firstName

setFirstName

```
public void setFirstName(java.lang.String value)
```

Set the value of firstName

getLastName

```
public java.lang.String getLastName()
```

Get the value of lastName

Returns:
the value of lastName

setLastName

```
public void setLastName(java.lang.String value)
```

Set the value of lastName

getUserName

```
public java.lang.String getUserName()
```

Get the value of userName

Returns:
the value of userName

setUserName

```
public void setUserName(java.lang.String value)
```

Set the value of userName

getPassword

```
public java.lang.String getPassword()
```

Get the value of password

Returns:

the value of password

setPassword

```
public void setPassword(java.lang.String value)
```

Set the value of password

getEmail

```
public java.lang.String getEmail()
```

Get the value of email

Returns:

the value of email

setEmail

```
public void setEmail(java.lang.String value)
```

Set the value of email

getExpire

```
public java.util.Date getExpire()
```

Get the value of expire

Returns:

the value of expire

setExpire

```
public void setExpire(java.util.Date value)
```

Set the value of expire

getCreated

```
public java.util.Date getCreated()
```

Get the value of created

Returns:

the value of created

setCreated

```
public void setCreated(java.util.Date value)
```

Set the value of created

getStatus

```
public Status getStatus()
```

Get the value of status

Returns:

the value of status

setStatus

```
public void setStatus(Status value)
```

Set the value of status

getSettingsXML

```
public java.lang.String getSettingsXML()
```

Get the value of settingsXML

Returns:

the value of settingsXML

setSettingsXML

```
public void setSettingsXML(java.lang.String value)
```

Set the value of settingsXML

getRole

```
public java.lang.String getRole()
```

Get the value of role

Returns:

the value of role

setRole

```
public void setRole(java.lang.String value)
```

Set the value of role

getJoinDate

```
public java.util.Date getJoinDate()
```

Get the value of joinDate

Returns:

the value of joinDate

setJoinDate

```
public void setJoinDate(java.util.Date value)
```

Set the value of joinDate

getExitDate

```
public java.util.Date getExitDate()
```

Get the value of exitDate

Returns:

the value of exitDate

setExitDate

```
public void setExitDate(java.util.Date value)
```

Set the value of exitDate

getType

```
public int getType()
```

Get the value of type

Returns:

the value of type

setType

```
public void setType(int value)
```

Set the value of type

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

fi.helsinki.cs.vaativa.model

Enum User.UserStatus

java.lang.Object

└ java.lang.Enum<[User.UserStatus](#)>

└ fi.helsinki.cs.vaativa.model.User.UserStatus

All Implemented Interfaces:

java.io.Serializable, java.lang.Comparable<[User.UserStatus](#)>

Enclosing class:

[User](#)

```
public static enum User.UserStatus  
extends java.lang.Enum<User.UserStatus>
```

Kelvolliset arvot Käyttäjän tilalle: ACTIVE= Käyttäjän tunnus on käytössä INACTIVE= ylläpitäjä on jäädyttänyt tunnuksen AWAITING_AUTHORIZATION= tunnus odottaa ylläpitäjän hyväksyntää

Enum Constant Summary

[ACTIVE](#)

[AWAITING_AUTHORIZATION](#)

[INACTIVE](#)

Method Summary

static User.UserStatus	valueOf (java.lang.String name) Returns the enum constant of this type with the specified name.
static User.UserStatus []	values () Returns an array containing the constants of this enum type, in the order they're declared.

Methods inherited from class java.lang.Enum

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

```
finalize, getClass, notify, notifyAll, wait, wait, wait
```

Enum Constant Detail

ACTIVE

```
public static final User.UserStatus ACTIVE
```

AWAITING_AUTHORIZATION

```
public static final User.UserStatus AWAITING_AUTHORIZATION
```

INACTIVE

```
public static final User.UserStatus INACTIVE
```

Method Detail

values

```
public static final User.UserStatus[] values()
```

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```
for(User.UserStatus c : User.UserStatus.values())  
    System.out.println(c);
```

Returns:

an array containing the constants of this enum type, in the order they're declared

valueOf

```
public static User.UserStatus valueOf(java.lang.String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

`name` - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`java.lang.IllegalArgumentException` - if this enum type has no constant with the specified name

[Package](#) [Class](#) [Use Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [ENUM CONSTANTS](#) | [FIELD](#) | [METHOD](#)

DETAIL: [ENUM CONSTANTS](#) | [FIELD](#) | [METHOD](#)
