

Määrittelydokumentti

ilpo-ryhmä

Helsinki 7.3.2005

Ohjelmistotuotantoprojekti

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

Kurssi

581260 Ohjelmistotuotantoprojekti (6 ov)

Projektiryhm

Arttu Eloranta
Risto Kuusterä
Antti Pulkkinen
Tommi Rajala
Tuomo Saarinen
Jari Suominen

Ohjaaja

Topias Marttila

Asiakas

Anni Rytönen

Vastuhenkilö

Juha Taina

Tekniset asiantuntijat

Juhani Haavisto
Harri Laine
Turjo Tuohiniemi

Kotisivu

<http://www.cs.helsinki.fi/group/ilpo>

Versiohistoria

Versio	Päiväys	Tehdyt muutokset
0.9	14.2.2005	Ensimmäinen versio, dokumentin rakenne lukossa.
1.0	20.2.2005	Draft: Yksityiskohdat yhä auki
1.5	25.2.2005	Korjattu Draft: Rärkeimpiä epäjohdonmukaisuuksia korjattu
2.0	25.2.2005	Asiakkaalle validoitavaksi tarjottava versio
2.1	7.3.2005	Korjauksilla validoitavaksi tarjottava versio

Sisältö

1 Johdanto	1
1.1 Taustaa	1
1.2 Termit ja määritelmät	1
1.3 Dokumentin rakenne	3
2 Yleisarkkitehtuuri	3
2.1 Järjestelmän yleiskuvaus	3
2.2 Järjestelmän tekninen kuvaus	4
2.3 Yleinen arkkitehtuurikuvaus	4
3 Vaatimukset	7
3.1 Toiminnalliset vaatimukset	7
3.2 Laadulliset vaatimukset	8
3.3 Suunnittelua rajoittavat vaatimukset	8
4 Toiminnot	9
4.1 Alkutoimenpiteet	9
4.2 Järjestelytoiminnot	11
4.3 Lopputoimenpiteet	18
5 Ulkoiset liittymät	19
5.1 Tietokannan kuvaus	19
5.2 Ryhmittelyn tallettaminen tietokantaan	24
5.3 Rajapinta Desmondiin	24
6 Rajoitteet suunnittelulle ja toteutukselle	25

Liitteet

1 Kyttöliittymä demo

1 Johdanto

Joillain kursseilla, kuten Ohjelmistotuotantoprojekti, Tieteellinen kirjoittaminen ja Opettajatuutorointi, opiskelijat jaetaan pienryhmiin ilmoittautumisten yhteydessä annettujen taustatietojen ja ryhmätoiveiden mukaan. Tavoitteena ryhmäjoossa on toisaalta muodostaa mahdollisimman tarkoituksenmukaisia, tasakokoisia ja/tai täysiä ryhmiä, ja toisaalta täyttää opiskelijoiden esittämiä toiveita niin paljon kuin mahdollista. Tarkoituksenmukaisuus voi tarkoittaa kurssista riippuen erilaisia asioita. Esimerkiksi tieteellisen kirjoittamisen kurssilla samaan ryhmään kuuluvien opiskelijoiden erikoistumisalueiden tulisi olla yhteneviä. Toisaalta ohjelmistotuotantoprojektissa samaan ryhmään on valittava eri tavoin suuntautuneita henkilöitä. Tällä hetkellä vastuuhenkilö tekee tämän jaottelun käsin paperilla.

1.1 Taustaa

Kursseilla Ohjelmistotuotantoprojekti, Tieteellinen kirjoittaminen ja Opettajatuutorointi on keskenään erilaiset sisällölliset ja hallinnolliset vaatimukset. Esimerkiksi Ohjelmistotuotantoprojektissa ryhmän koon on oltava 4-6 opiskelijaa. Kurssilla on pakolliset esitietovaatimukset, ja joissain aiheissa saattaa olla lisävaatimuksiakin suoritetuista esitietokursseista. Opettajatuutorointiin ilmoittautuvat opiskelijat ovat eri vaiheissa opinnoissaan ja suorittavat niitä eri tahtiin. Osa opiskelijoista tietää, mihin haluaa suuntautua ja suuntautumisvaihtoehdot. Ryhmät pyritään muodostamaan siten, että opiskelu olisi opiskelijoille mielekäästä.

Ilpo on Helsingin yliopiston tietojenkäsittelytieteen laitoksella toteutettava projekti. Projekti toteutetaan osana opintojaksoa Ohjelmistotuotantoprojekti. Projektin tarkoitus on tuottaa sovellus, joka auttaa opintojaksojen Opettajatuutorointi, Tieteellinen kirjoittaminen ja Ohjelmistotuotantoprojekti vastuuhenkilöitä ryhmäjakojen teossa. Kaikki projektin aikana tuotettava materiaali tulee saataville ryhmän kotisivuilla:

<http://www.cs.helsinki.fi/group/home/ilpo>

Tämän dokumentin tarkoitus on esitellä määritellyt tuotettavasta sovelluksesta tuomalla julki kokonaiskuva sovelluksesta, sen rajapinnoista ja ulkoisista liittymistä. Lisäksi karotetaan asiakkaan esittämät vaatimukset joiden pohjalta määritellään toteutettavat toiminnot.

1.2 Termit ja määritelmät

Desmond (<http://ilmo.cs.helsinki.fi/desmond/>): Aiempien ohjelmistotuotantoprojektien tuottama opintojensuunnittelujärjestelmä. Desmondin www-selaimessa toimivalla käyttöliittymällä opiskelijat voivat suunnitella opintojaan lukukausittain. Desmondissa on myös muita toimintoja, jotka auttavat esimerkiksi tuutoroinnissa, mutta ne ovat projektin kannalta epärelevanttejä.

Ilmo (<http://ilmo.cs.helsinki.fi/>): Kurssi-ilmoittautumisjärjestelmä opiskelijoille. Järjes-

telmä on tarkoitettu tietojenkäsittelytieteen laitoksen kursseille ilmoittautumiseen. Ilmolla ilmoitaudutaan myös matematiikan kursseille, mutta sillä ei ole tämän projektin kannalta juuri mitään merkitystä.

Ilpo: Sovelluksen toteuttavan ohjelmistotuotantoprojektiryhmän ja lopullisen järjestelmän nimi.

JAR (Java Archive file format): JAR:n avulla voidaan pakata useita tiedostoja yhteen JAR-tiedostoon. Tyypillisesti JAR-tiedosto sisältää luokkatiedostoja sekä sovelluksiin ja sovelmiin liittyviä lisäresursseja.

JDBC (<http://java.sun.com/products/jdbc/>): Sun Microsystemsin JDBC-teknologia on Java-kielelle toteutettu sovellusrajapinta (API, Application Programming Interface) joka tarjoaa liityntämahdollisuuden useiden eri valmistajien relaatiotietokantoihin ja mm. laskentataulukoihin (spreadsheets).

J rjestyskriteeri: Järjestyskriteerit ovat Ilmo-lomakkeen yksittäisiä vastauksia: esimerkiksi suuntautumislinja, kieli, esitiedot, opiskelijan tila.

Kurki: Laitoksen henkilökunnan käyttöön tarkoitettu kurssikirjanpitojärjestelmä johon merkitään mm. opiskelijoiden tekemät laskuharjoitukset. Käyttää samaa tietokantaa kuin Ilmo.

Kurssi: Kurssi on opintojakson ilmentymä. Kurssi määritellään erikseen kyseessä olevalle periodille, lukukaudelle ja -vuodelle.

K ytt j : Ilpolla ryhmittelyä suorittava henkilö.

Lukitus: Yhden opiskelijan kiinnittäminen pysyvästi tiettyyn ryhmään siten, että opiskelija ei näy enää muissa ryhmissä.

Oodi (<http://www.helsinki.fi/weboodi/>): Opiskelun ja opetuksen tuen tietojärjestelmä. Oodista löytyvät muun muassa opiskelijoiden opintosuoritukset.

Opintojakso: Opintojakso määrittelee opetettavalle aihepiirille nimen ja laajuuden sekä muut yleiset luentokerrasta riippumattomat tiedot.

Opiskelija: Kurssille ilmoittautunut henkilö, jota sijoitetaan ilpolla ryhmiin.

PL/SQL (Procedural Language/SQL): PL/SQL on laajennus SQL-kyselykieleen. Se lisää perus-SQL:ään proseduraalisten ohjelmointikielten rakenteita, jolloin tuloksena on perus-SQL:ää ilmaisuvoimaisempi rakenteellinen kieli.

Ryhmittely (Ryhmiiin jako): Opiskelijoiden jakaminen ryhmiin.

Ryhm : Kokonaisuus mihin ilpolla siirretään opiskelijoita. Ryhmässä on opiskelijoita ja henkilökunnan jäseniä.

Ryhm lukitus: Ryhmän lukitseminen siten, että kaikki ryhmässä olevat opiskelijat eivät enää näy muissa ryhmissä.

Sarjallistuva olio: Ohjelmointikielen olio, jonka tila voidaan konvertoida tavuvirraksi (ja toisinpäin). Käytännössä tämä tarkoittaa, että koko olio voidaan esimerkiksi talentaa kiintolevyille myöhempää käyttöä varten. Sarjallistuvia olioita on ainakin Java-ohjelmointikielessä.

Sis inen tietorakenne: Olorakenne, joka on pelkästään ilpon sisäisessä käytössä ja mi- hin on tallennettu tietoja.

Tavallinen TKTL:n pöyt kone: Käyttöjärjestelmänä Linux graafisella ikkunointiympä- ristöllä, näyttö 1280x1024 resoluutiolla sekä vähintään 1.5 GHz suoritin ja vähintään 512Mt keskusmuistia.

Tk_oph: Etuliite, jota käytetään viitattaessa Ilmo-järjestelmän tietokantatauluihin Des- mondista käsin. Käytännössä "Tk_oph = Ilmo".

TKTL: Helsingin yliopiston tietojenkäsittelytieteen laitos.

Tulosryhm : Järjestyskriteerin mukaisesti tehty järjestely jakaa opiskelijat tulosryhmiin.

1.3 Dokumentin rakenne

Dokumentin alkupuolella luvuissa kaksi ja kolme luodaan yleinen kuva järjestelmän ra- kenteesta sekä määritellään asiakkaan vaatimukset järjestelmälle. Tämän jälkeen mää- ritellään yksityiskohtaisesti järjestelmän toiminnot asiakkaan vaatimuksien pohjalta sekä määritellään tarvittavat yhteydet ulkoisiin järjestelmiin. Varsinkin rajapinnat ulkoisiin jär- jestelmiin pyritään määrittelemään mahdollisimman yksityiskohtaisesti. Koska ulkoisten järjestelmien tarjoamista palveluista ei ole vielä tässä vaiheessa tarkkoja tietoja ryhmän ulkopuolisista tahoista johtuen, voivat nämä määrittelyt muuttua projektin edetessä. Ilpon käyttöliittymän yleispiirteet esitellään erillisessä dokumentissaan.

2 Yleisarkkitehtuuri

Luku sisältää kuvauksen järjestelmän toteutusympäristöstä ja ulkoisista liittymistä.

2.1 Järjestelmän yleiskuvaus

Ohjelmiston on tarkoitus olla työkalu, joka auttaa kurssin vastuuhenkilöä opiskelijoiden profiloinnissa ja ryhmäjaossa. Ohjelman avulla voidaan tarkistaa kurssille ilmoittautuneiden esitiedot Oodista missä tahansa vaiheessa ryhmiinjaottelua. Ohjelman käyttäjä voi ryhmitellä opiskelijoita määriteltyjen kriteerien ja prioriteettien perusteella. Käyttäjä voi siirrellä opiskelijoita ryhmästä toiseen. Käyttäjä voi myös lukita opiskelijoita ryhmiin. Jotta ohjelman käyttäjä pystyy tuottamaan toimivan ryhmäjaon, ohjelma pyrkii esittämään opiskelijoiden vastaukset, toiveet ja mahdolliset esitietopuutteet mahdollisimman tiiviis- sä ja selkeässä muodossa. Ohjelma on suunniteltu käytettäväksi pääasiallisesti kurssien Tieteellinen kirjoittaminen, Ohjelmistotuotantoprojekti ja Opettajatuutorointi ilmoittau- tumisten yhteydessä.

Ohjelmiston avulla käyttäjä voi jaotella kursseille ilmoittautujia heidän toiveidensa mu- kaisesti ryhmiin esimerkiksi kielen tai suuntautumislinjan mukaan. Hän voi karsia joukosta henkilöt, joiden ennakkotiedoissa on puutteita tai jättää puutteet huomioimatta. Tyypilli-

sesti ryhmäjaossa on tavoitteena muodostaa mahdollisimman homogeenisia, tasakokoisia ja/tai täysiä ryhmiä. Ryhmät pyritään muodostamaan siten, että opiskelu olisi opiskelijoille mielekästä. Ohjelmisto käyttää opiskelijoiden ja kurssien tietojen hankinnassa hyväkseen Ilmo ja Desmondia, jotka ovat yhteydessä Oodiin. Käyttäjä voi luoda useita erilaisia jaotteluja, tallentaa niitä missä vaiheessa tahansa sekä ladata tallennettuja ryhmäjakoa. Saatuaan tyydyttävän ryhmäjaon aikaiseksi, käyttäjä pystyy jäädyttämään ryhmäjaon ja kirjaamaan sen Ilmoon. Ohjelmalla voidaan myös tuottaa ryhmäjaottelu HTML-koodina liitettäväksi esimerkiksi laitoksen valmiiseen sivupohjaan, jolloin ryhmäjaon saa helposti esitettyä kurssin sivulla.

2.2 Järjestelmän tekninen kuvaus

Ohjelma toimii Java-sovelluksena graafisella käyttöliittymällä terästettynä. Laitevaatimuksena on, että sovellus toimii tavallisessa TKTL:n pöytäkoneessa (ks. termit). Tämä tarkoittaa implisiittisesti, että järjestelmää voidaan käyttää ainoastaan TKTL:n sisäverkosta käsin. Käyttöjärjestelmällä ei pitäisi olla merkitystä, kunhan sille löytyy sopiva Java-virtuaalikone. Java-versiona käytetään JDK 5.0:aa (eli J2SE SDK 1.5.0) ja Desmond-tietokantaan ollaan yhteydessä JDBC-rajapinnan kautta. Työn alla olevat ryhmittelyt tallennetaan levyille sarjallistuvina olioina ja valmiit ryhmittelyt Ilmo-tietokantaan jo olemassa olevan PL/SQL-skriptin avulla.

Sisäänkirjautuminen ilpoon tapahtuu Desmond/Ilmo-tietokannassa olevan "profiili-käyttäjän salasanalla. "Profiili-käyttäjä on ilpon tarpeita varten luotu käyttäjä, jolla on oikeus tehdä ryhmittelyjä ja nähdä ne tietokannan taulut, jotka ovat ryhmittelyn kannalta tarpeellisia. Käyttäjän nimi saattaa olla toinen valmiissa järjestelmässä. Käyttäjätunnus ja salasana ovat yhteisiä kaikille ilpon käyttäjille. Tässä ratkaisussa Oraclen tietokannanhallintajärjestelmä hoitaa ilpo-järjestelmän käyttäjien autentikoinnin. Tämä on riittävä tapa autentikoida käyttäjät, sillä järjestelmän kurssien vastuuhenkilöistä koostuva käyttäjäkunta on suppea ja nämä henkilöt ovat lähtökohtaisesti luotettavia.

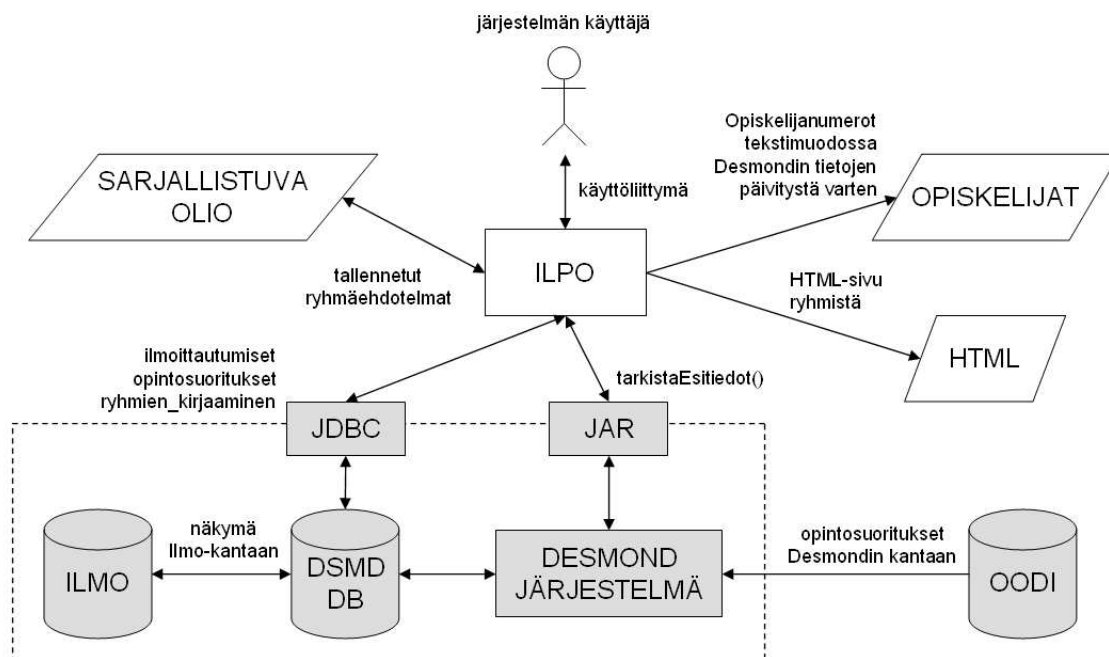
2.3 Yleinen arkkitehtuurikuvaus

Edellisten ohjelmistotuotantoprojektien tuottamaa opintojensuunnittelujärjestelmää, Desmondia, hyödynnetään kahdella eri tavalla; ilpo on yhteydessä suoraan Desmondin tietokantaan, jonka kautta se näkee paitsi Oodi-järjestelmästä Desmondin tietokantaan noudetut opiskelijoiden opintosuoritukset, myös Ilmo-järjestelmään tallennetut kurssi-ilmoitautumiset. Desmondin tietokannasta on myös näkymä Ilmo-kannasta löytyviin kurssin hallinnollisiin tietoihin, kuten luennoitsijoihin ja ryhmäjakoihin. Desmondin tietokannan Ilmo-näkymän kautta päivitetään valmiit ryhmäjaot Ilmo-tietokantaan. Ilpo-projektia varten käytössä on myös testikanta, joka sisältää tietyn valikoiman Ilmon ja Desmondin tietokantatauluja. Todellisuudessa Desmondilla ja Ilmolla ei ole omaa tietokantaa vaan niiden taulut sijaitsevat yhdessä suuressa tietokannassa. Käsitteiden selventämiseksi kutsumme näitä "virtuaalisia tietokantoja" tavallisiksi tietokannoiksi.

Desmondin valmiita palveluita hyödynnetään opiskelijoiden esitietovaatimusten tarkasta-

miseen. Tätä varten projektiryhmälle toimitetaan valmis JAR-tiedosto, jonka avulla palvelua voidaan hyödyntää.

Kuvassa 1 on järjestelmän tietovirtoja selventävä kaavio. Tikku-ukko kuvaa järjestelmän käyttäjää, joka antaa sekä saa tietoa ilpo-järjestelmästä graafisen käyttöliittymän avustuksella. Ennen järjestelmän käytön aloittamista, käyttäjän tulee syöttää validi käyttäjätunnus ja salasana. Valkoiset objektit kuvaavat järjestelmän osia ja tulosteita, jotka ilpo-ryhmä ja -järjestelmä tulevat toteuttamaan. Harmaat objektit kuvaavat sovellusrajapintoja, tietokantoja ja toista järjestelmää.



Kuva 1: Selventävä kaavio

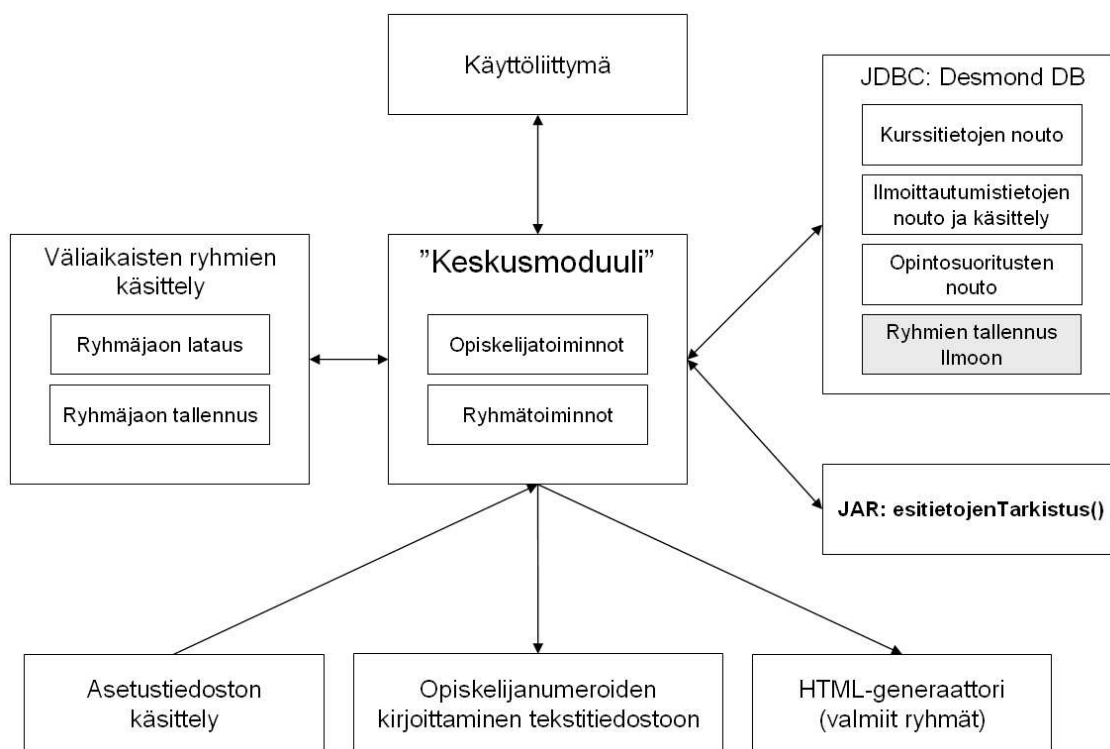
Ilpo-järjestelmä on kuvattu valkoisena suorakulmiona. Valkoiset suunnikkaan muotoiset objektit kuvaavat järjestelmästä saatavaa tai siihen tuotavaa tietoa. Opiskelijat-suunnikas kuvaa tekstitiedostoa, johon järjestelmä kirjoittaa niiden opiskelijoiden opiskelijanumerot, joiden suoritustiedot halutaan päivitettävän Oodista Desmondiin. HTML-suunnikas kuvaa tekstimuotoista HTML-tiedostoa, johon järjestelmä kirjoittaa valmiit ryhmittelyt. HTML-tiedostosta selviää ainakin ryhmän aiheen nimi sekä ohjaajan ja osallistujien nimet. Sarjallistuva olio -suunnikas kuvaa binääritiedostoa, johon järjestelmä tallettaa väliaikaiset ryhmittelyvaihtoehdot ja josta ne ladataan. Sarjallistuvia olioita voi olla useita jotta järjestelmän käyttäjä pystyy helposti tallentamaan ja lataamaan useita eri ryhmittelyvaihtoehtoja.

Harmaat suorakulmiot JDBC ja JAR kuvaavat järjestelmän käyttämiä sovellusrajapintoja. Järjestelmä hakee JDBC-sovellusrajapinnan kautta tietoa Desmondin tietokannasta (DSMD DB). JAR-pakettina toimitettavan sovellusrajapinnan tarkoituksena on tarkastaa Desmondista opiskelijoiden esitietovaatimusten täyttyminen profiloitavan kurssin suhteen. Desmond-suorakulmio saa ilpolta palvelukutsuja sovellusrajapinnan kautta (JAR-

paketti). Desmond lukee ja kirjoittaa oman tietokantansa tietoja. Lisäksi Desmond saadaan noutamaan Oodi-tietokannasta ajantasaiset opiskelijoiden kurssisuoritustiedot. Tämä onnistunee ilpon tuottaman opiskelijanumerolistan avulla, mutta varsinainen tietojen päivitys on rajattu projektin ulkopuolelle teknisistä rajoitteista johtuen.

Harmaat ympyrälieriöt kuvaavat tietokantoja. Oodi-tietokanta tuottaa tietoa Desmond-järjestelmälle. Desmondin tietokanta saa Desmond-ohjelmalta sekä JDBC-rajapinnalta luku- ja kirjoituskyselyjä. Lisäksi Desmond-tietokannasta on näkymä Ilmo-tietokantaan. Tämän näkymän kautta pystytään JDBC:n avulla lukemaan useita Ilmo-tietokantatauluja sekä kirjoittamaan tiettyihin niistä.

Kuvassa 2 on ilpo-järjestelmän karkeaa sisäistä rakennetta selventävä kaavio. Järjestelmässä on eräänlainen keskusmoduuli, jonka sisälle kuuluvat opiskelija- ja ryhmätoiminnot. Lisäksi keskusmoduuli toimii välikätenä eri moduuleiden välillä. Opiskelijatoimintoihin kuuluvat mm. opiskelijoiden tietojen lisääminen, muokkaaminen ja poistaminen järjestelmän sisäisestä tietorakenteesta. Ryhmätoimintoihin kuuluvat mm. opiskelijoiden lisääminen ryhmään ja poistaminen ryhmästä, ryhmien lisääminen ja ryhmien lukitseminen järjestelmän sisäisessä tietorakenteesta.



Kuva 2: Sisäinen rakenne

Käyttöliittymä-moduuli keskustelee käyttäjän pyyntöjen ja muiden moduulien toimien perusteella keskusmoduulin kanssa. Käyttäjän lähettämät pyynnöt ohjataan keskusmoduulin kautta muille moduuleille. Vastaavasti käyttäjä saa muiden moduulien lähettämää tietoa keskusmoduulilta.

Väliaikaisten ryhmien käsittely -moduuli sisältää väliaikaisten ryhmäjakojen tallennuksen

ja latauksen. Operaatiot tehdään käyttäjän pyynnöstä ja tiedostot tallennetaan käyttäjän valitsemaan paikkaan.

JDBC Desmond DB -moduuli sisältää tarvittavat operaatiot tietojen lukemiseksi ja kirjoittamiseksi Desmondin tietokantaan. Kurssitietojen nouto noutaa tarvittavat profiloitavaan kurssiin liittyvät tiedot Desmond-tietokannan Ilmo-näkymän kautta. Ilmoittautumistietojen nouto ja käsittely noutaa oppilaan tekemät kurssi-ilmoittautumistiedot Desmond-tietokannan Ilmo-näkymän kautta. Opintosuoritustennouto noutaa opiskelijoiden kurssi-suoritukset Desmond-järjestelmästä. Ryhmien tallennus Ilmoon tallentaa valmiit ryhmitelut ilmoon käyttämällä valmista PL/SQL-skriptiä.

JAR-paketti sisältää metodin esitietojen tarkistamiseen. Paketti on kuvattu tarkemmin luvussa ulkoiset liittymät.

Asetustiedoston käsittely -moduuli sisältää ilpo-järjestelmän käyttöliittymän ja raken-teiden mukauttamisen profiloitavien kurssien (ohjelmistotuotantoprojekti, opettajatuoto-rointi ja tieteellinen kirjoittaminen) tarpeiden mukaan. Eri kurssit sisältävät esimerkiksi eri määrän kurssi-ilmoittautumisen yhteydessä vastattavia kenttiä. Opiskelijanumeroiden kirjoittaminen tekstitiedostoon -moduuli kirjoittaa tekstitiedoston niiden opiskelijoiden opiskelijanumeroista, joiden suoritustiedot halutaan päivitettävän Oodista Desmondiin. HTML-generaattori-moduulin avulla luodaan valmiiksi ryhmitellyistä kursseista HTML-tiedosto.

3 Vaatimukset

Tässä luvussa kuvataan asiakkaan esittämät vaatimukset sovellukselle. Nämä eivät ole ohjelmiston yksittäisten toimintojen kuvauksia, vaan sovelluksen tasolla tapahtuvia il-miöitä, jotka järjestelijä haluaa tämän ohjelmiston avulla suorittaa. Vaatimukset on jaot-teltu kolmeen luokkaan. Luokat ovat Toiminnalliset vaatimukset, Laadulliset vaatimukset ja Suunnittelua rajoittavat vaatimukset. Jokaiselle vaatimukselle on myös yksikäsitteinen tunniste.

3.1 Toiminnalliset vaatimukset

Tässä kuvataan asiat, joita ohjelmiston avulla pitää pystyä suorittamaan. Vaatimukset ja niiden tunnistenumerot ovat seuraavat:

- V1** Käyttäjän pitää voida tarkastella eri opiskelijoiden tietoja.
- V2** Käyttäjän pitää voida siirrellä, lukita ja poistaa sekä opiskelijoita ryhmistä, että ko-konaisia ryhmiä.
- V3** Käyttäjän pitää voida järjestellä opiskelijoita ilmoittautumistietojen ja taustatietojen perusteella.

- V4 Käyttäjän pitää voida tallentaa ryhmäjaottelu missä vaiheessa tahansa ja hänellä voi olla useita ryhmäjaotteluja.
- V5 Käyttäjän pitää voida avata uudelleen haluamansa tallennettu ryhmäjako.
- V6 Käyttäjällä pitää olla toiminto, jolla hän voi tuoda kurssille ilmoittautuneet, päivittää heidän tietojaan ja tuoda myöhässä Ilmoon lisätyt uudet ilmoittautujat.
- V7 Käyttäjän pitää saada tulostettua jaottelunsa HTML-muodossa.
- V8 Käyttäjän pitää voida tallentaa tekemänsä jaottelu Ilmo-tietokantaan.

3.2 Laadulliset vaatimukset

Laadullisia vaatimuksia voidaan tarkastella seuraavista näkökulmista: ohjelmiston nopeus tai suorituskyky, muistin tai levytilan käyttö, luotettavuus, käytettävyys, tietoturva, tuettavat standardit yms.

- V9 Sovelluksen täytyy olla selkeämpi ja helppokäyttöisempi, kuin aikaisemmin käytössä olleet paperiset versiot
- V10 Sovelluksen täytyy olla tietoturvallinen, koska käsitellään opiskelijoiden henkilötietoja.
- V11 Rajapintasovellusten kuten Desmondin muutokset eivät saa vaikuttaa toimivuuteen. Tässä oletetaan, että muutokset tehtäisiin olio-ohjelmointisuunnittelujen oppeja käyttäen. Tietokannan rakennemuutokset saattavat vaikuttaa toimivuuteen.
- V12 Järjestelmän sisäiset toiminnot eivät saa normaali käytössä aiheuttaa yli neljän sekunnin viiveitä. Haut tietokannasta ja sinne talletukset eivät kuulu tehokkuuden mittauksen piiriin.

3.3 Suunnittelua rajoittavat vaatimukset

Tästä löytyvät vaatimukset, jotka asettavat rajoituksia suunnittelulle ja toteutukselle.

- V13 Sovelluksen pitää saada kurseja koskevaa dataa ja pystyä päivittämään muokattua dataa Ilmosta Desmondin näkymien kautta.
- V14 Sovelluksen pitää saada opiskelijoita koskevaa dataa Oodista Desmondin kautta.
- V15 Koska Desmond ei osaa käsitellä korvattuja opintoja ja sen tiedot opiskelijoiden suorittamista kurseista eivät ole aina ajan tasalla, on järjestelmän tarjottava käyttäjälleen toiminto, jolla tämä voi merkitä ilpon suorittamattomiksi merkitsemiä kurseja suoritetuiksi.

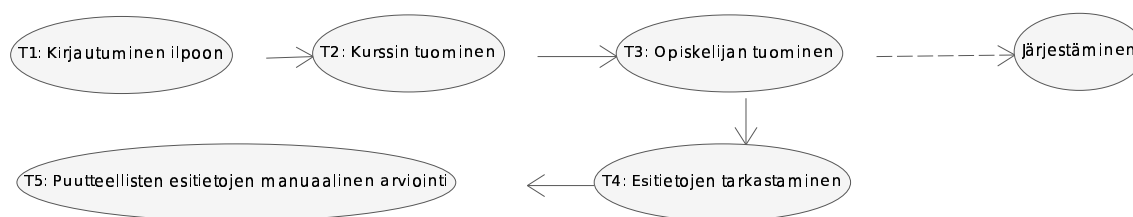
V16 Sovelluksen on tarjottava myös manuaalinen tapa opiskelijoiden opiskelijanumeroiden syöttämiseen, jotta niiden opiskelijoiden opintosuorituksia voidaan tarkastella jotka eivät ole kyseistä tietoa Ilmo-järjestelmään ilmoittaneet.

4 Toiminnot

Ilpoon toteutettavat toiminnot on listattu seuraavassa. Toiminnoista on listattu vaiheet ja variaatiot. Lisäksi ne on priorisoitu siten, että tärkeimmät toiminnot ovat merkitty prioriteetiksi yksi. Prioriteetilla yksi merkityt toiminnot ovat ohjelman toiminnan kannalta välttämättömiä. Prioriteetin kaksi toiminnot ovat ohjelman käytettävyyden kannalta oleellisia. Kolmanneksi priorisoidut toiminnot ovat käyttökelpoisia lisäominaisuuksia. Ryhmän tavoitteena on toteuttaa kaikki luvussa luetellut toiminnot. Jos prioriteetin kolme toimintoja jää toteuttamatta, voidaan aikaansaautua järjestelmää pitää hyvänä. Ainoastaan prioriteetin yksi implementoivaa järjestelmää voidaan pitää tyydyttävänä.

4.1 Alkutoimenpiteet

Ennen varsinaista järjestämistä suoritettavien toimenpiteiden suhteita toisiinsa on selvennetty kuvassa 3.



Kuva 3: Alkutoimenpiteiden välisiä yhteyksiä

T1 Kirjautuminen ilpoon

Kuvaus: Käyttäjä kirjautuu ilpoon

Tavoite: Taata valituille henkilöille mahdollisuus päästä käyttämään ilpoa ja estää ilpon luvaton käyttö.

1 Käyttäjä syöttää tietokannan salasanan.

2 Ilpo tarkistaa, että salasana on oikea.

Prioriteetti: 1

Jäljitettävyys: V10

T2 Kurssin tuominen

Kuvaus: Käyttäjä valitsee kurssin ennen varsinaisen työskentelyn aloittamista.

Tavoite: Kaikki ilmoittautumisen yhteydet saadut tiedot on järkevästi järjestelty käyttäjän ulottuville ryhmiin jakamista varten.

Vaiheet:

1 Käyttäjä valitsee käsiteltävän kurssin, jonka tiedot haetaan Ilmosta.

2 Ilpo hakee kurssin opiskelijoiden tiedot kannasta (T3).

Prioriteetti: 1

Jäljitettävyyys: V6, V11

T3 Opiskelijan tuominen

Kuvaus: Toiminnolla haetaan opiskelijan tiedot Ilmosta.

Tavoite: Kaikki ilmoittautumisen yhteydessä saadut tiedot on järkevästi järjestelty käyttäjän ulottuville ryhmiin jakamista varten.

Vaiheet:

1 Ilpo tietää opiskelijan, jonka tiedot tuodaan.

2 Ilpo tarkistaa, että ilmoittautumisjärjestelmästä saadut tiedot ovat oikeellisessa muodossa.

2a Jos osa ilmoittautumisjärjestelmästä saadusta tiedosta ei ole oikeellisessa muodossa.

2a1 Käyttäjä korjaa syötteen oikeelliseksi.

3 Ilpo tarkistaa opiskelijan esitiedot(T3).

Variaatiot:

1 Jos sisäisessä järjestelmässä on opiskelija, ei häntä tuoda toiseen kertaan Ilmosta.

2 Jos opiskelija on poistettu Ilmosta, häntä ei poisteta sisäisestä rakenteesta, mutta hänet merkitään poistetuksi.

Prioriteetti: 1

Jäljitettävyyys: V6, V11, V13, V14

T4 Esitietojen tarkastaminen

Kuvaus: Tarkistetaan kurssille ilmoittautuneiden opiskelijoiden esitiedot, jotka kyseiselle kurssille vaaditaan.

Tavoite: Saada selville ketkä opiskelijoista eivät ole suorittaneet vaadittavia esitietoja, ja mitkä esitiedot heiltä puuttuvat.

Vaiheet:

1 Ilpo tarkistaa Oodista/Desmondista, ovatko suoritukset kunnossa.

1.b Jos suoritukset eivät ole kunnossa, ilpo merkitsee opiskelijan ongelmatapaukseksi.

1.c Jos suoritukset eivät aiemmin olleet kunnossa, mutta ovat T3:n uudelleenajon myötä kunnostautuneet, ilpo merkitsee opiskelijan entiseksi ongelmatapaukseksi.

Variaatiot:

1 Mikäli opiskelija on erikseen merkitty ongelmattomaksi esitietojen suhteen (T3), ei ilpo yliaja tätä määrittelyä, mikäli tämä toiminto suoritetaan myöhemmin uudestaan.

Prioriteetti: 2

Jäljitettävyyys: V3, V9, V11, V13, V14

T5 Puutteellisten esitietojen manuaalinen arviointi

Kuvaus: Esitietojen tarkistamisessa ongelmatapaukseksi luokitellun henkilön tilanteen tarkempi analyysi.

Tavoite: Pitää huoli siitä, etteivät viime hetkellä arvosteltavat kurssit, jotka opiskelijalta vaaditaan osallistumiseen, estä hänen kurssille osallistumistaan.

Vaiheet:

1 Ilpo kertoo, mitkä esitiedot puuttuvat opiskelijalta.

2.a Jos käyttäjä tietää opiskelijan läpipääsyn olevan mahdollinen, merkitsee hän kurssin ilpoon oletusarvoisesti läpäistyksi.

3.b Jos käyttäjä tietää läpipääsyn mahdottomaksi, käyttäjä poistaa opiskelijan kursilta.

Prioriteetti: 1

Jäljitettävyyys: V6, V9, V11, V13, V14, V15, V16

4.2 Järjestelytoiminnot

Järjestelyn aikana käytettävien toimintojen välisiä yhteyksiä on esitetty kuvassa 4.

T6 Ryhmän luominen

Kuvaus: Luodaan uusi ryhmä.

Tavoite: Ryhmä on luotu.

Vaiheet:

1 Käyttäjä luo uuden ryhmän.

Prioriteetti: 1

Jäljitettävyyys: V2

T7 Ryhmän nimeäminen

Kuvaus: Nimetään ryhmä.

Tavoite: Annetaan ryhmälle kuvaava nimi, jotta ryhmän erottaa muista ryhmistä.

Vaiheet:

1 Käyttäjä valitsee ryhmän.

2 Käyttäjä antaa ryhmälle nimen.

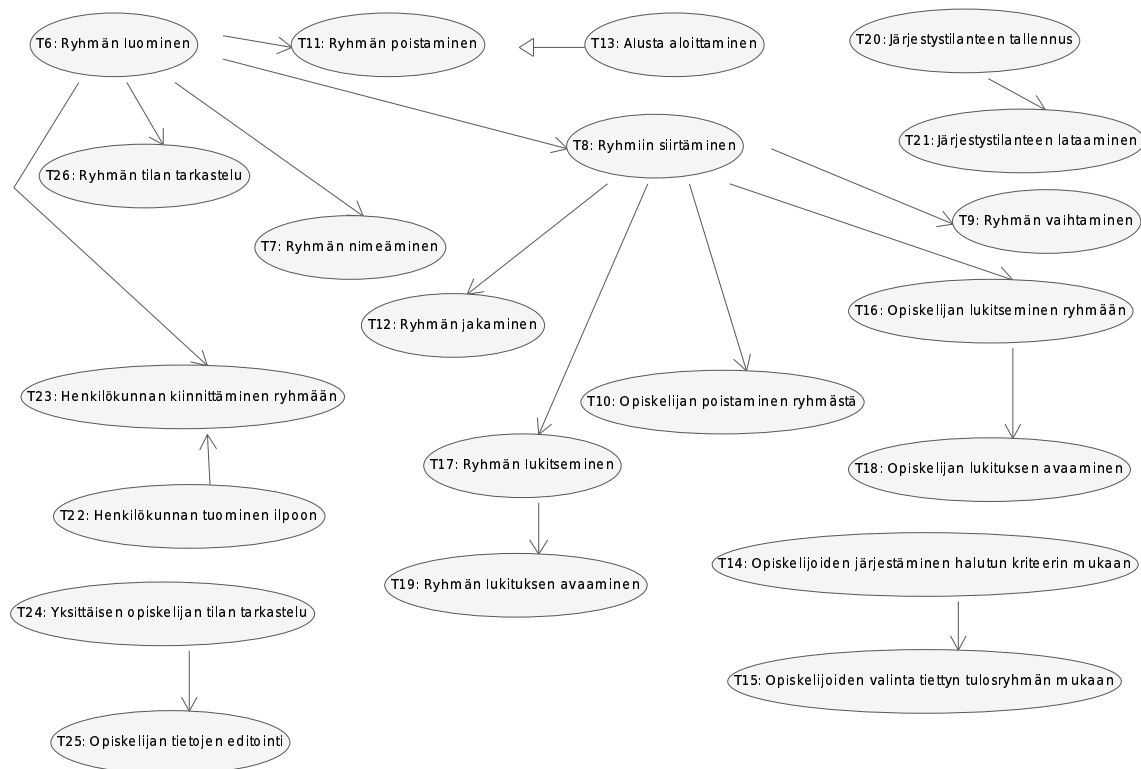
3 Ilpo kirjoittaa ryhmän nimen sisäiseen tietorakenteeseen.

Variaatio:

1 Käyttäjä muuttaa ryhmän nimeä.

Prioriteetti: 2

Jäljitettävyys: V2, V9



Kuva 4: Järjestelytoimenpiteiden välisiä yhteyksiä

T8 Ryhmiin siirtäminen

Kuvaus: Opiskelija sijoitetaan ryhmään.

Tavoite: Opiskelija on sopivassa ryhmässä.

Vaiheet:

1 Käyttäjä valitsee opiskelijan.

2 Käyttäjä valitsee ryhmän.

2.a Jos valittu ryhmä on uusi, luodaan uusi ryhmä (T6).

3 Ilpo asettaa valitun opiskelijan valittuun ryhmään.

Variaatiot:

1 Käyttäjä valitsee useita opiskelijoita, jotka sijoitetaan samaan ryhmään tällä toiminnolla.

Prioriteetti: 1 (vaiheen 2.a, sekä variaatio 1:n prioriteetti 2)

Jäljitettävyys: V2, V9

T9 Ryhmän vaihtaminen

Kuvaus: Opiskelija siirretään ryhmästä toiseen.

Tavoite: Opiskelija on sijoitettu uuteen ryhmään ja poistettu aikaisemmasta ryhmästä (move).

Vaiheet:

1 Käyttäjä valitsee opiskelijan ryhmästä.

2 Käyttäjä valitaan toisen ryhmän.

2.a Jos toinen ryhmä on uusi luodaan uusi ryhmä (T6).

Variaatiot:

1 Käyttäjä valitsee useita opiskelijoita, jotka sijoitetaan samaan ryhmään tällä toiminnolla.

Prioriteetti: 2 (variaatio 1:n prioriteetti 3)

Jäljitettävyys: V2, V9

T10 Opiskelijan poistaminen ryhmästä

Kuvaus: Poistetaan opiskelija ryhmästä

Tavoite: Opiskelija on poistettu ryhmästä

Vaiheet:

1 Käyttäjä valitsee opiskelijan.

2 Käyttäjä valitsee ryhmän, josta hänet poistetaan.

3 Ilpo poistaa opiskelijan ryhmästä.

Variaatiot

1 Suoritetaan vaiheet käänteisessä järjestyksessä.

Prioriteetti: 1

Jäljitettävyys: V2

T11 Ryhmän poistaminen

Kuvaus: Poistetaan ryhmä.

Tavoite: Ilposta on poistettu ryhmä.

Vaiheet:

1 Käyttäjä valitsee ryhmän.

2 Ilpo poistaa ryhmän.

Variaatiot:

1 Jos jokin ryhmässä olevista opiskelijoista on lukittu ryhmään, ilpo ei poista ryhmää.

Prioriteetti: 1 (variaatio 1:n prioriteetti 2)

Jäljitettävyys: V2

T12 Ryhmän jakaminen

Kuvaus: Ryhmä jossa on opiskelijoita jaetaan kahdeksi ryhmäksi.

Tavoite: Ryhmä on jaettu kahdeksi ryhmäksi.

Vaiheet:

1 Käyttäjä valitsee ryhmän.

2 Ilpo luo uuden ryhmän (T6).

3 Ilpo jättää lukitut opiskelijat alkuperäiseen ryhmään.

4 Loput opiskelijat ilpo jakaa alkuperäiseen ja uuteen ryhmään tasaisesti.

Prioriteetti: 3

Jäljitettävyys: V2, V9

T13 Alusta aloittaminen.

Kuvaus: Aloitetaan ryhmiin jakaminen alusta.

Tavoite: Tämänhetkistä ryhmäjakoja paremman jaon löytäminen.

Vaiheet:

1 Käyttäjä aloittaa kurssin ryhmittelyn alusta.

2 Ilpo poistaa kaikki ryhmittelyt.

Prioriteetti: 3

Jäljitettävyys: V4, V9

T14 Opiskelijoiden järjestäminen valitun kriteerin mukaan.

Kuvaus: Opiskelijat järjestetään valitun kriteerin mukaan ryhmiin jaottelun helpottamiseksi.

Tavoite: Opiskelijat on jaoteltu.

Vaiheet:

1 Käyttäjä valitsee halutun kriteerin.

2 Ilpo järjestää opiskelijat kriteerin mukaisesti.

Huomautus:

1 Kriteerit ovat ilmolomakkeen yksittäisiä vastauksia, suuntautumisinja, kieli, esitiedot, opiskelijan tila...

Prioriteetti: 1

Jäljitettävyys: V3, V9

T15 Opiskelijoiden valinta tiettyjen kriteerien perusteella.

Kuvaus: Valitaan kaikki haluttuihin tulosryhmiin kuuluvat opiskelijat.

Tavoite: Nähdä niiden opiskelijoiden tiedot, jotka täyttävät tietyt ehdot.

Vaiheet:

1 Käyttäjä valitsee yhden tai useamman tulosryhmän.

Prioriteetti: 2

Jäljitettävyyys: V1, V3, V9

T16 Opiskelijan lukitseminen ryhmään.

Kuvaus: Lukitaan opiskelija ryhmään.

Tavoite: Lukita opiskelija ryhmään ja piilottaa hänet kaikista muista ryhmistä, joihin hänet on sijoitettu.

Vaiheet:

1 Käyttäjä valitsee opiskelijan.

2 Käyttäjä valitsee ryhmän.

3 Ilpo lukitsee opiskelijan ryhmään.

Variaatiot:

1 Vaiheitten järjestys voi vaihdella.

Prioriteetti: 2

Jäljitettävyyys: V2, V9

T17 Ryhmän lukitseminen.

Kuvaus: Lukitaan koko ryhmä.

Tavoite: Kaikki yhteen ryhmään valitut henkilöt lukittuvat tähän ryhmään ja poistuvat muista ryhmistä.

Vaiheet:

1 Käyttäjä valitsee ryhmän.

2 Ilpo lukitsee ryhmän.

Prioriteetti: 2

Jäljitettävyyys: V2

T18 Opiskelijan lukituksen avaaminen.

Kuvaus: Poistetaan opiskelijan lukitus.

Tavoite: Ryhmään lukittu opiskelija ei ole enää lukittu.

Vaiheet:

1 Käyttäjä valitsee opiskelijan.

2 Ilpo poistaa lukituksen.

Variaatiot:

1 Jos koko ryhmä on lukittu (T17), ei ilpo poista yksittäisen opiskelijan lukitusta.

Prioriteetti: 2 (variaatio 1:n prioriteetti 3)

Jäljitettävyys: V2

T19 Ryhmän lukituksen avaaminen.

Kuvaus: Poistetaan ryhmän lukitus.

Tavoite: Lukittu ryhmä ei ole enää lukittu.

Vaiheet:

1 Käyttäjä valitsee lukitun ryhmä.

2 Ilpo poistaa lukituksen.

Variaatio:

1 Jos joku ryhmän opiskelija on lukittu henkilökohtaisesti ryhmään, ryhmän lukituksen poistaminen ei poista tätä lukitusta.

Prioriteetti: 2 (variaatio 1:n prioriteetti 3)

Jäljitettävyys: V2

T20 Järjestystilanteen tallennus.

Kuvaus: Tallennetaan järjestystilanne myöhempää käyttöä varten.

Tavoite: Jatkaa järjestämistä mahdollisesti myöhemmin.

Vaiheet:

1 Käyttäjä valitsee kurssin tallentamisen.

2 Ilpo tallentaa kurssia edustavan sisäisen tietorakenteen sarjallistuvana oliona levyille.

Prioriteetti: 2

Jäljitettävyys: V4, V9

T21 Järjestystilanteen lataaminen.

Kuvaus: Ladataan aiemmin tallennettu järjestystilanne.

Tavoite: Aiemman tilanteen palautus.

Vaiheet:

1 Käyttäjä valitsee aiemmin tallennetun kurssin tilanteen.

2 Ilpo tuo kurssia edustavat sarjallistuvan olion levyiltä sisäiseksi tietorakenteeksi.

Prioriteetti: 2

Jäljitettävyys: V5, V9

T22 Henkilökunnan tuominen ilpoon.

Kuvaus: Tuodaan Ilmosta henkilökunnan jäsenet, jotka on jo linkitetty ryhmäkohdaisesti toisiinsa.

Tavoite: Tuoda tietokannasta kaikki kurssille ilmoitetut henkilökunnan jäsenet.

Vaiheet:

1 Ilpo tuo henkilökunnan jäsenet.

Prioriteetti: 2

Jäljitettävyyys: V9

T23 Henkilökunnan kiinnittäminen ryhmään.

Kuvaus: Kiinnitetään henkilökunta ryhmään.

Tavoite: Assistenttien/Vastuuhenkilöiden kiinnittäminen.

Vaiheet:

1 Käyttäjä valitsee ryhmähenkilökunnan.

2 Käyttäjä valitsee ryhmän.

3 Ilpo liittää henkilökunnan ryhmään.

Variaatiot:

1 Vaiheiden 1 ja 2 järjestys voi vaihdella.

Prioriteetti: 1

Jäljitettävyyys: V9

T24 Yksittäisen opiskelijan tilan tarkastelu.

Kuvaus: Tarkastella yksittäisen opiskelijan tilaa.

Tavoite: Nähdä yksittäisen opiskelijan tila.

Vaiheet:

1 Käyttäjä valitsee opiskelijan.

2 Ilpo näyttää valitun opiskelijan kaikki tiedot.

Variaatiot:

1 Käyttäjä valitsee usean opiskelijan, joiden tila näytetään.

Prioriteetti: 1 (variaatio 1:n prioriteetti 2)

Jäljitettävyyys: V1, V14

T25 Opiskelijoiden tietojen editointi

Kuvaus: Muutetaan opiskelijan antamia vastauksia ryhmäjaon aikana.

Tavoite: Opiskelijoiden tiedot ovat järkeviä ja oikeellisia.

Vaiheet:

- 1 Käyttäjä valitsee opiskelijan.
- 2 Käyttäjä valitsee tiedon.
- 3 Käyttäjä muuttaa tiedon.
- 4 Ilpo muuttaa tiedon sisäisessä tietorakenteessa.

Prioriteetti: 2

Jäljitettävyys: V1, V6, V14

T26 Ryhmän tilan tarkastelu.

Kuvaus: Tarkastella yksittäisen ryhmän tilaa.

Tavoite: Nähdä yksittäisen ryhmän tila.

Vaiheet:

- 1 Käyttäjä valitsee ryhmän.
- 2 Ilpo näyttää valitun ryhmän tiedot.

Prioriteetti: 2

Jäljitettävyys: V1, V3, V9

4.3 Lopputoimenpiteet

T27 HTML-tuloste

Kuvaus: Tulostetaan ryhmäjako HTML-sivuna.

Tavoite: Ryhmäjako paperi/web-muodossa.

Vaiheet:

- 1 Ilpo tulostaa ryhmäjaon tiedostoon.

Tuloste:

Toiminto tuottaa tuloksenaan osan HTML-tiedostosta, joka talletetaan käyttäjän valitsemalla nimellä levyille. Tiedoston sisältö on vastaavanlainen, kuin sivulla <http://www.cs.helsinki.fi/~ilpo/ryhmajako.html>, rajoittuen kuitenkin kommenttien `<!-- header loppuu -->` ja `<!-- trailer alkaa -->` väliseen osuuteen.

Prioriteetti: 2

Jäljitettävyys: V7, V9

T28 Ryhmäjaon jäädyttäminen.

Kuvaus: Ryhmäjaon valmistuttua työskentelyn lopettaminen ja jaon vieminen kantaan.

Tavoite: Ryhmäjako on kirjattu ilmoon.

Vaiheet:

- 1 Ilpo kirjoittaa ryhmäjaon Ilmoon.

Tuloste:

Toiminnon suorittamisen jälkeen Ilmo-tietokannasta löytyy tehty ryhmittely. Järjestelmä on suunniteltu siten, että toiminto suoritetaan ainoastaan kerran yhden kurssin ryhmittelyä kohti.

Prioriteetti: 1

Jäljitettävyyys: V8, V9, V10, V11

5 Ulkoiset liittymät

Ilpo-järjestelmä on yhteydessä sekä Desmond-järjestelmään, että sen tietokantaan. Osa Ilmo-tietokannan tauluista on käytettävissä Desmond-tietokannassa. Desmond-järjestelmää käytetään kurssille vaadittavien esitietojen tarkastamiseen ja sen tietokannasta noudetaan lähes kaikki profiloinnissa tarvittavat tiedot.

5.1 Tietokannan kuvaus

Tietokannasta, jota ilpo-järjestelmä käyttää, löytyy visualisointi kuvassa 5.

Koska Ilmo-tietokanta on sama kuin opetushenkilökunnan kurssikirjanpitojärjestelmän, se sisältää myös lähes kaikki profiloitavan kurssin hallinnollisista tiedoista. Ainoa tieto, mitä tietokannasta ei löydy, on kurssin sisäisten ryhmien nimet, jotka jäävät siten käyttäjän täytettäväksi.

Seuraavassa on kuvattu ne tietokannasta löytyvät taulut, jota ilpo-järjestelmä käyttää. Ainoastaan ilpon käyttämät kentät on lueteltu. Kahdesta tietokannasta olevat taulut ovat nähtävissä yhdessä loogisessa tietokannassa. Tk_opho-etuliitteellä varustetut taulut sijaitsevat fyysisesti Ilmo-tietokannassa ja desmond-etuliitteiset taulut Desmond-tietokannassa, johon ilpo on siis yhteydessä.

tk_opho.opintojakso

Taulussa on pysyvät opintojakson tiedot. Kurssikoodia käytetään yksilöimään opintojaksot ja taulusta löytyvä suomenkielisillä nimellä tarkennetaan Desmondista löytyviä opintosuoritusmerkintöjä.

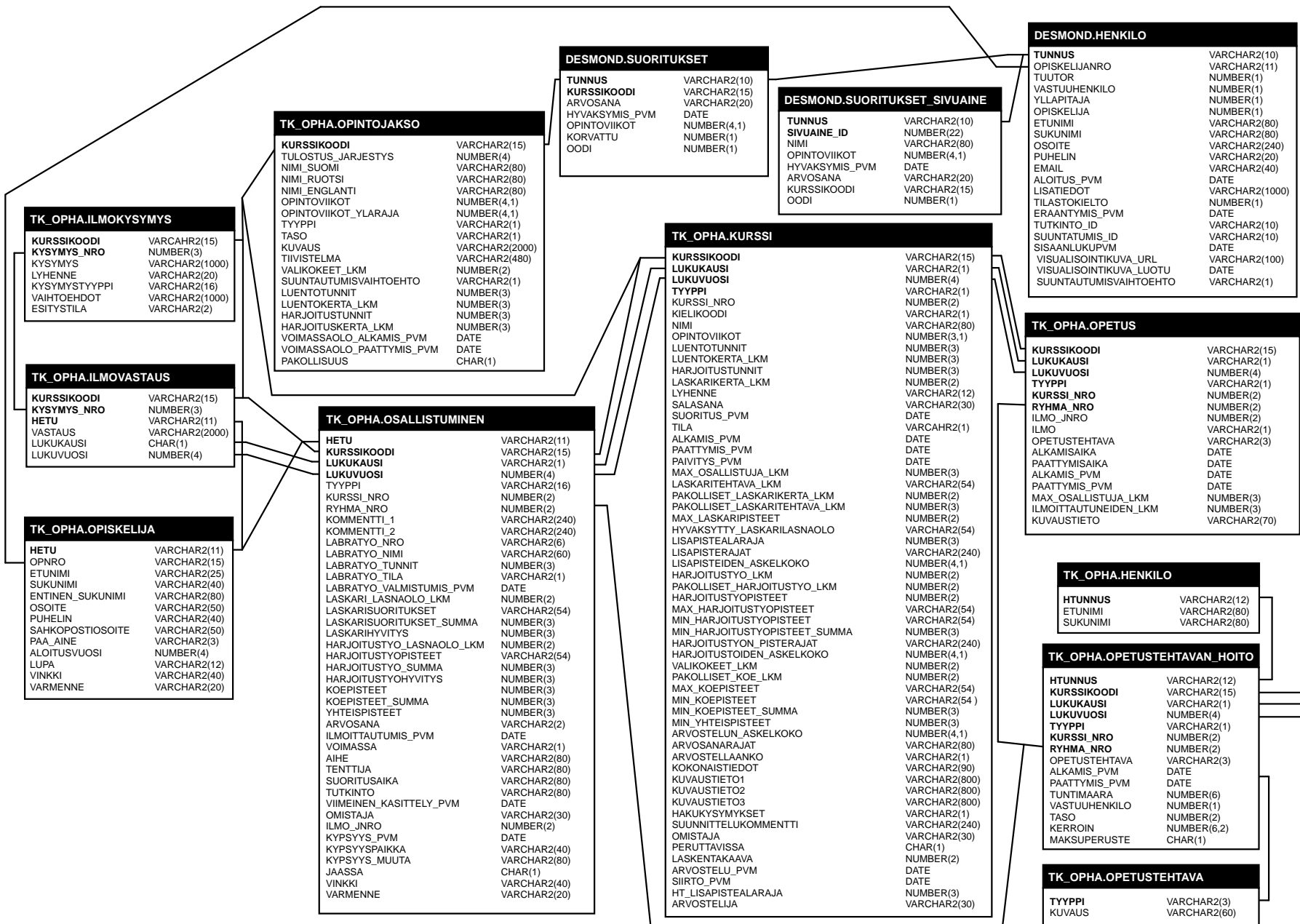
kurssikoodi (varchar2(15)): Opintojakson yksilöivä tunnus.

nimi_suomi (varchar2(80)): Opintojakson suomenkielinen nimi.

tk_opho.kurssi

Tietyllä lukukaudella luennoitava opintojakson ilmentymä. Liittyy opintojaksoon kurssikoodin välityksellä. Sisältää tiedot lukukaudesta ja -vuodesta sekä kurssin nimestä ja

Kuva 5: Käytävät tietokantataulut



kurssin alkamis- ja loppumispäivämääristä. Lisäksi sisältää järjestysnumeron, jolla erotellaan saman lukukauden aikana luennoitavat saman opintojakson ilmentymät.

kurssikoodi (varchar2(15)): Viite opintojaksoon, jonka ilmentymä kyseinen kurssi on.

lukukausi (varchar2(1)): Lukukausi, jolloin kurssi pidetään. Lukukaudet on koodattu siten, että kevät on *K*, syksy *S* ja kesä *V*.

lukuvuosi (number(4)): Vuosi, jolloin kurssi järjestetään.

kurssi_nro (number(2)): Tunnus, jolla erotellaan saman lukukauden aikana pidettävät kurssit toisistaan. Mikäli esimerkiksi saman lukukauden aikana järjestetään kaksi erillistä ohjelmistotuotantoprojektia on ensimmäisen *kurssi_nro* 1 ja toisen 2.

tk_ophi.opetus

Sisältää tiedot yhdestä kurssin pienryhmästä. Liittyy kurssiin kurssikoodin, kurssin järjestysnumeron ja lukukauden sekä -vuoden välityksellä. *Opetus*-taulun *ilmoittautuneiden_lkm*-kenttää päivitetään tietokannassa olevalla skriptillä.

kurssikoodi, lukukausi, lukuvuosi, kurssi_nro: Toimivat yhdessä viitteenä kurssiin, jonka pienryhmä on kyseessä.

ryhma_nro (number(2)): Pienryhmän järjestysnumero kurssin sisällä.

ilmoittautuneiden_lkm (number(3)): Pienryhmässä olevien opiskelijoiden lukumäärä.

tk_ophi.opiskelija

Opiskelijan tiedot Ilmo-järjestelmässä: henkilötunnus, opiskelijanumero, nimi, sähköpostiosoite, pääaine ja aloitusvuosi.

Koska opiskelijat joutuvat tarkastamaan nämä osana Ilmo-järjestelmään sisäänkirjautumista, ilpo käyttää tästä taulusta löytyviä henkilötietoja desmond.henkilo-taulusta löytyvien tietojen sijaan. Desmond-järjestelmästä löytyvät opiskelijakohtaiset tiedot voidaan yhdistää Ilmo-järjestelmän vastaaviin opiskelijanumeron avulla.

hetu (varchar2(11)): Opiskelijan henkilötunnus, käytetään Ilmo-järjestelmässä avaimena.

opnro (varchar2(25)): Opiskelijan opiskelijanumero, liittää Ilmo-järjestelmän opiskelijat Desmond-järjestelmässä oleviin opintosuorituksiin.

etunimi (varchar2(25)): Opiskelijan etunimi.

sukunimi (varchar2(40)): Opiskelijan sukunimi.

entinen_sukunimi (varchar2(80)): Opiskelijan entinen sukunimi, käyttötarve kyseenalainen.

sahkopostiosoite (varchar2(50)): Opiskelijan sähköpostiosoite.

paa_aine (varchar2(3)): Kolmimerkkinen lyhenne opiskelijan pääaineesta, viittaa *paa_aine*-tauluun. Koska *paa_aine*-taulua ei ole käytettävissä, ilpossa opiskelijoiden pää-aineet esitetään lyheenteinä.

aloitusvuosi (number(4)): Vuosi, jolloin opiskelija on aloittanut yliopisto-opintonsa.

tk_oph.osallistuminen

Ilmo-järjestelmä lisää tauluun rivin aina opiskelijan ilmoittautuessa kurssille. Taulun avulla opiskelija yhdistetään kurssin ryhmään henkilötunnusta, kurssikoodia, lukukautta ja -vuotta sekä ryhmänumeroa käyttäen. Kun ilmoittautumiset on ryhmitelty ilpon avulla, järjestelmä päivittää taulun rivejä siten, että opiskelijat siirretään "ylivuoto-ryhmästä varsinaisiin ryhmiin. Tämä päivitys tapahtuu tietokannassa olevaa skriptiä kutsuen.

hetu (varchar2(11)): Viittaa kurssille osallistuvaan opiskelijaan.

kurssikoodi, lukukausi, lukuvuosi, kurssi_nro, ryhmä_nro: Toimivat viitteenä kurssin ryhmään, johon opiskelija osallistuu.

voimassa (number(1)): Tieto siitä, onko ilmoittautuminen vielä voimassa. Voimassa olevilla ilmoittautumisilla arvo on *K* ja peruilla *E*.

tk_oph.ilmokysymys

Ilmo-järjestelmässä ollut kysymys. Avain on kurssikoodin ja kysymysnumeron yhdistelmä.

kurssikoodi (varchar2(15)): Viite opintojaksoon, johon ilmoittautumisessa on kysymystä käytetty.

kysymys_nro (number(3)): Kysymyksen numero Ilmo-järjestelmässä.

kysymys (varchar2(20)): Ilmo-järjestelmässä esillä ollut kysymys.

lyhenne (varchar2(16)): Kysymyksen lyhenne.

kysymystyyppi (varchar2(16)): Kysymystyyppi, mahdolliset vaihtoehdot *text* (vapaakenttä) ja *radio* ("radiobutton", arvot 1 (valittu) ja 2 (ei valittu)).

vaihtoehdot (varchar2(1000)): Kysymyksen vastausvaihtoehdot, eri vaihtoehdot on erotettu toisistaan pilkulla (,).

esitystila (varchar2(2)): Tieto siitä, onko kysymys nähtävillä ilmoittautumisjärjestelmässä.

tk_oph.ilmovastaus

Opiskelijan vastaus Ilmo-järjestelmän kysymykseen. Avaimena toimii henkilötunnuksen, kurssikoodin ja kysymyksen numeron yhdistelmä.

kurssikoodi, lukuvuosi, lukukausi: Viite kurssiin, jonka ilmoittautumiseen vastaus liittyy.

hetu (varchar2(11)): Viite kysymykseen vastanneeseen opiskelijaan.

kysymys_nro (number(3)): Kysymyksen numero Ilmo-järjestelmässä.

vastaus (varchar2(2000)): Opiskelijan antama vastaus kysymykseen.

tk_oph.henkilo

Ilmo-järjestelmässä olevat tiedot laitoksen henkilökuntaan kuuluvasta henkilöstä. Avaimena toimiva htunnus muodostetaan henkilön nimestä. Ainoa taulussa oleva tieto on henkilön nimi.

htunnus (varchar2(12)): Henkilökunnan jäsenen yksilöivä tunnus.

etunimet (varchar2(80)): Henkilökunnan jäsenen etunimet.

sukunimi (varchar2(80)): Henkilökunnan jäsenen sukunimi.

tk_oph.opetustehtavan_hoito

Liittää henkilökunnan jäsenen opetus-tauluun. Sisältää myös tiedot opetustehtävän laadusta viitteenä opetustehtava-tauluun. Taulun avulla ryhmiin liitetään niiden asiakkaat, valvojat, ohjaajat sekä muut henkilöt.

htunnus (varchar2(12)): Viittaa henkilökunnan jäsenen

kurssikoodi, lukukausi, lukuvuosi, kurssi_nro, ryhmä_nro: Toimivat viitteenä kurssin ryhmään.

opetustehtava (varchar2(2)): Opetustehtävän laadun lyhenne. Tarkempaa tietoa löytyy *opetustehtava*-taulusta.

tk_oph.opetustehtava

Erityyppisten opetustehtävien kuvaukset, esimerkiksi asiakas, valvoja tai ohjaaja.

tyyppi (varchar2(3)): Viite *opetustehtavan_hoito*-taulun *opetustehtava*-kentässä olevaan opetustehtävän lyhenteeseen.

kuvaus (varchar2(60)): Opetustehtävän kuvaus, esimerkiksi ohjaaja tai valvoja.

desmond.henkilo

Desmond-järjestelmään talletettu opiskelija. Avaimena toimii käyttäjätunnus yliopiston ATK-järjestelmiin. Voidaan yhdistää vastaaviin Ilmo-järjestelmän opiskelijoihin opiskelijanumeron avulla, jolloin Desmond järjestelmässä olevat opintosuoritukset voidaan liittää ilmoittautuneisiin opiskelijoihin.

tunnus (varchar2(10)): Käyttäjätunnus yliopiston ATK-järjestelmiin

opiskelijanro (varchar2(11)): Opiskelijan opiskelijanumero

desmond.suoritukset

Opiskelijan suorittamat tietojenkäsittelytieteen kurssit, sekä niiden arvosanat. Liitettävissä Ilmo-järjestelmän kurssitietoihin kurssikoodin avulla. Liittyvät desmond.henkilo-

tauluun käyttäjätunnuksen avulla.

tunnus (varchar2(10)): Käyttäjätunnus yliopiston ATK-järjestelmiin.

kurssikoodi (varchar2(10)): Viite Ilmo-kannassa olevaan opintojaksoon.

arvosana (varchar2(20)): Opintosuorituksen arvosana.

hyvaksymis_pvm (date): Opintojakson hyväksymispäivämäärä.

opintoviikot (number(4,1)): Opintojakson laajuus.

desmond.suoritukset_sivuaine

Opiskelijan suorittamat sivuainekurssit, sekä niiden arvosanat. Liittyvät *desmond.henkilo*-tauluun käyttäjätunnuksen avulla.

tunnus (varchar2(10)): Käyttäjätunnus yliopiston ATK-järjestelmiin.

kurssikoodi (varchar2(10)): Suoritetun opintojakson yksilöivä tunnus.

sivuaine_id (number(22)): Yksilöllinen sivuaineen tunniste.

nimi (varchar2(80)): Opintojakson nimi.

arvosana (varchar2(20)): Opintosuorituksen arvosana.

hyvaksymis_pvm (date): Opintojakson hyväksymispäivämäärä.

opintoviikot (number(4,1)): Opintojakson laajuus.

5.2 Ryhmittelyn tallettaminen tietokantaan

Valmiiden ryhmien tallettaminen tietokantaan tapahtuu tietokannassa olevalla skriptillä, jonka tarkempi rakenne selvitetään suunnitteluvaiheessa. Skriptin ideana on kuitenkin muuttaa opiskelijoiden *osallistuminen*-taulussa olevaa *ryhma_nro*-kenttää, samalla joudutaan päivittämään ainakin *opetus*-taulussa olevia opiskelijoiden lukumäärätietoja.

5.3 Rajapinta Desmondiin

Ilpo käyttää Desmond-järjestelmän tarjoamaa palvelua esitietovaatimusten tarkastamiseen. Rajapintana järjestelmään toimii projektille toimitettava JAR-paketti joka sisältää tähän soveltuvan metodin. Esitietojen tarkastaminen vaatii sen, että ilpolla on käytettävissä ajantasaiset tiedot opiskelijoiden opintosuorituksista. Nykyisellään Desmond ei pysty tätä takaamaan, joten siihen on lisättävä komponentti, jonka avulla ilpo voi pyytää tarvittavien tietojen lisäämistä ja päivittämistä.

tarkastaEsitiedot(suoritukset:Collection, kurssi:String): Collection

Parametreina annetaan tarkastettavan kurssin kurssikoodi (merkkijonona), sekä Collection-rajapinnan toteuttava joukko, joka sisältää tarkastettavan opiskelijan suorittamien kurssien koodit (merkkijonoina). Metodi palauttaa Collection-rajapinnan toteuttavan olion, joka sisältää niiden kurssien kurssikoodit (merkkijonoina), jotka kyseiseltä opiskelijalta puuttuvat, jotta hän voisi osallistua annetulle kurssille.

Parametrit:

Collection suoritukset: opiskelijan suorittamat kurssit merkkijonoina

String kurssi: kurssi merkkijonona, jonka esitiedot halutaan tarkastaa

Kurssitietojen päivitys

Jotta Desmondin tietokannassa olisi opiskelijoiden ajantaiset opintosuoritukset on ne noudettava Oodista. Tämä onnistunuu ainoastaan Ilmo-koneelta, joten päätteillä toimivan ilpo-järjestelmän on kyettävä listaamaan päivitettävien opiskelijoiden opiskelijanumerot ja tallettamaan listaus levyille tekstitiedostona, joka sisältää yhden opiskelijanumeron riviä kohti. Tuotettua tiedostoa hyväksikäyttäen on mahdollista päivittää Desmondin tietokanta.

6 Rajoitteet suunnittelulle ja toteutukselle

Projektin käyttöön tarkoitettu testikanta on osittain toteutettu. Pienistä puutteista johtuen tietokantakuvausten täsmällinen määrittely on mahdotonta. Kuvausta tarkennetaan jatkossa.

Vastuuhenkilöt ryhmittelevät opiskelijat ainoastaan oman vastuukurssinsa puitteissa. Täten muiden kurssien vastuuhenkilöiden ei tule muokata muiden keskeneräisiä ryhmittelyjä. Keskeneräisten ryhmittelyjen usean käyttäjän samanaikainen muokkaaminen vaatisi vähintäänkin tiedostotason keskinäisen poissulkemisen. Tämä ei ole joustava tapa lukita tietoa usean käyttäjän ympäristössä. Joustavampien tapojen suunnittelu ja toteutus rajataan tämän projektin ulkopuolelle.

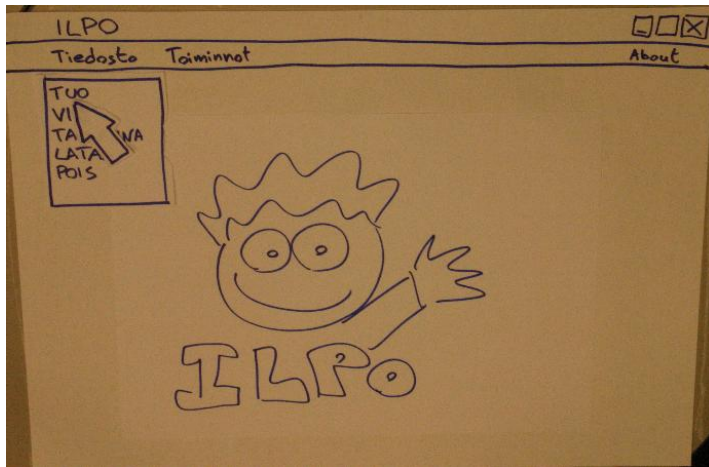
Liite 1

Käyttöliittymädemo

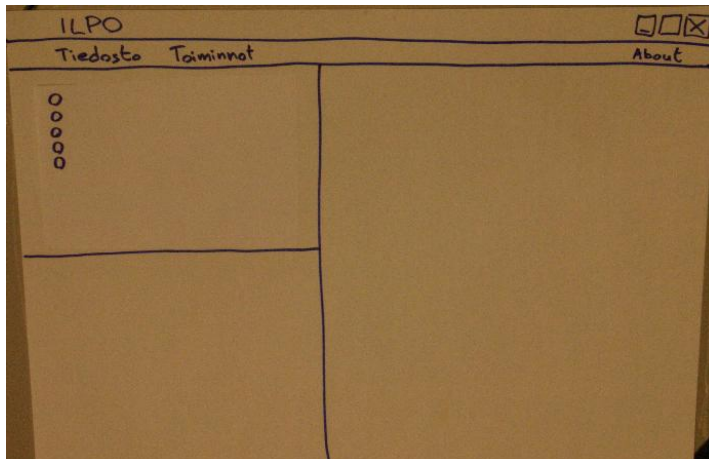
27.02.2005



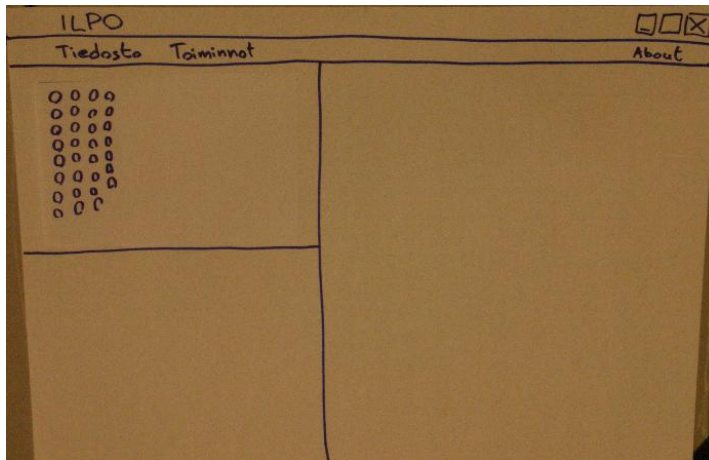
Ipolla on tarkoitus järjestellä opiskelijoita kurssin ryhmiin heidän ilmoittautumistietojensa perusteella.



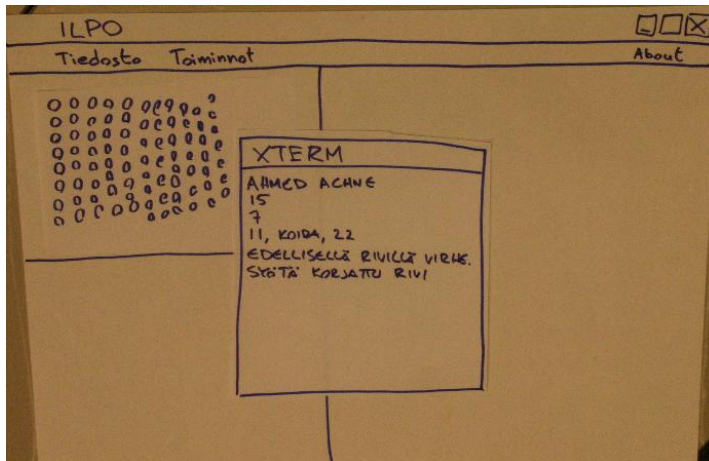
Aluksi käyttäjä valitsee valikosta kurssin tietojen tuomisen. Hän valitsee järjesteltävän opintojakson ja kurssin lukukauden.



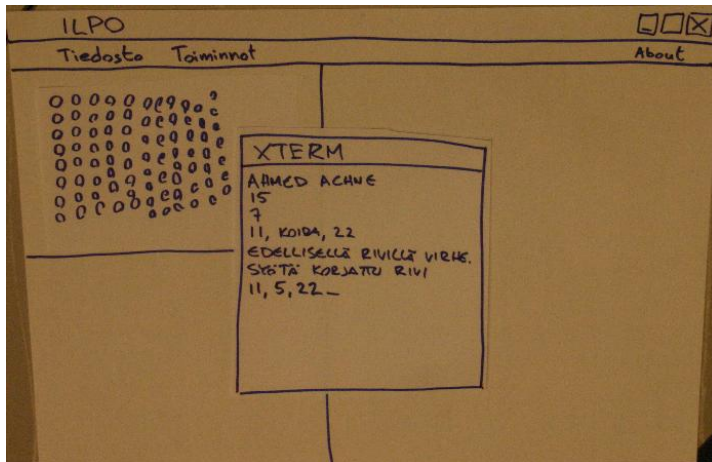
Ipo hakee kurssille ilmoittautuneita opiskelijoita ilmoittautumistiedot sisältävästä tietokannasta.



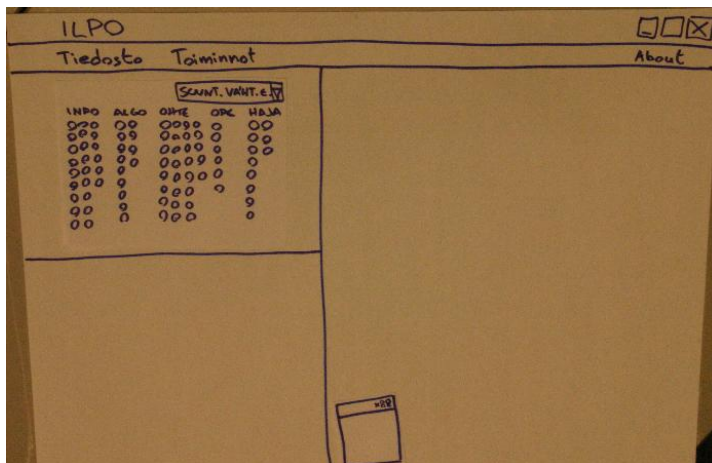
Ilpo hakee opiskelijoita edelleen.



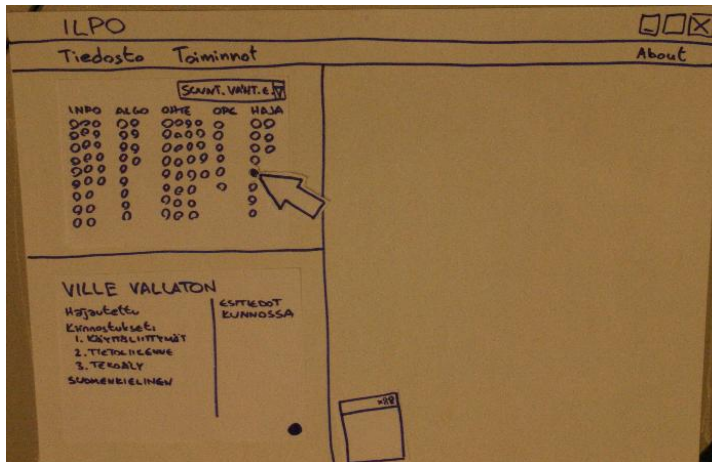
Ilpo havaitsi, että "Ahmed Achne"-niminen opiskelija on syöttänyt ilmoittautumistietonsa virheellisessä formaatissa ja pyytää nyt käyttäjää syöttämään virheellisen rivin uudelleen, että ilpo kykenee tulkitsemaan, mitä opiskelija on yrittänyt sanoa.



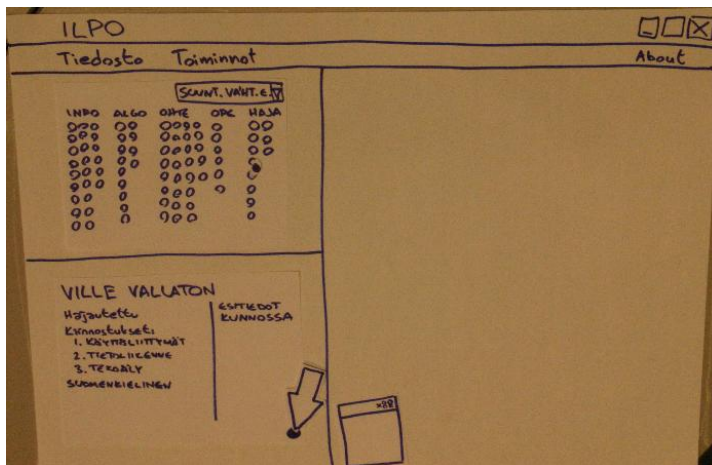
Käyttäjää korjaa opiskelijan virheellisesti syötetyn rivin poistamalla riviltä tekstin ja muuttamalla rivin pilkuin erotelluiksi numeroiksi, jonka ilpo osaa parsia.



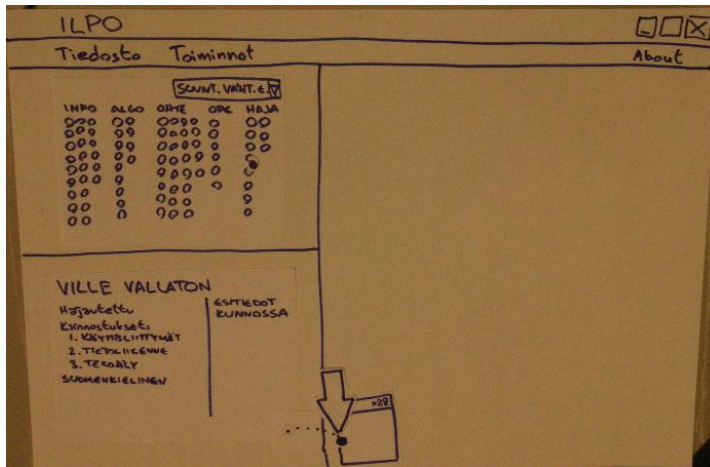
Ilo on saanut haettua kaikkien kurssille ilmoittautuneiden opiskelijoiden tiedot. Heidät on aluksi järjestetty vasemmassa ylänurkassa sijaitsevaan tilaan opiskelijoiden ilmoittaman suuntautumisvaihtoehdon mukaan.



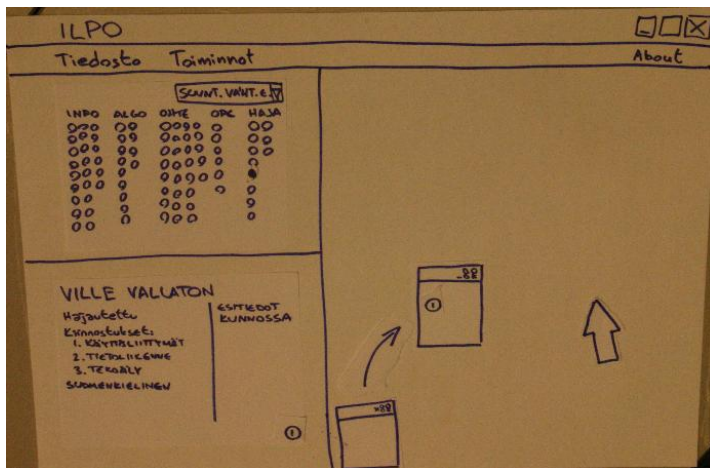
Käyttäjä valitsee yhden satunnaisen "Hajautettuihin järjestelmiin" suuntautuvan opiskelijan (Ville Vallaton) klikkaamalla opiskelijaa symboloivaa palloa ruudulla. Kaikki ko. opiskelijan tiedot (ilmoittautumisen yhteydessä annetut sekä muut eri järjestelmistä haetut) tulevat näkyviin vasemmassa alareunassa sijaitsevaan ruutuun.



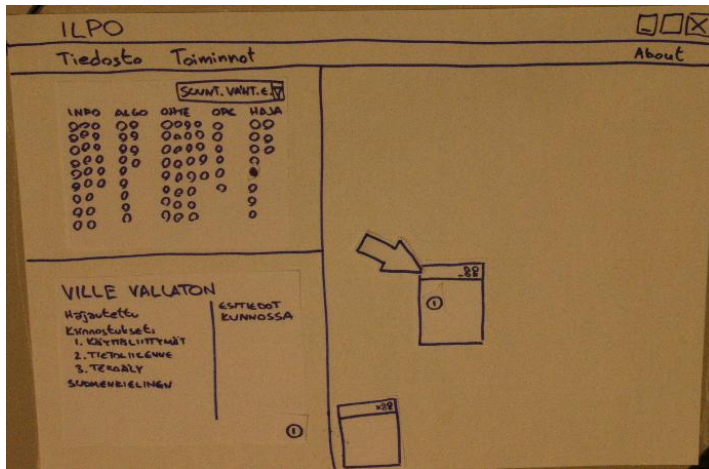
Käyttäjä haluaa siirtää Villen ryhmään. Villeä esittävä pallo on sekä ylä- että alaruudussa. Käyttäjä ottaa kiinni alaruudussa olevasta pallosta ...



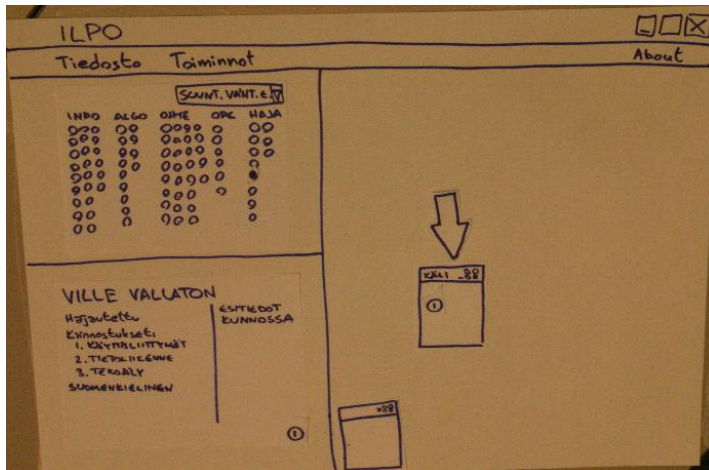
... raahaa pallon oikealla suurella ryhmätilassa sijaitsevaan tyhjään ryhmään ...



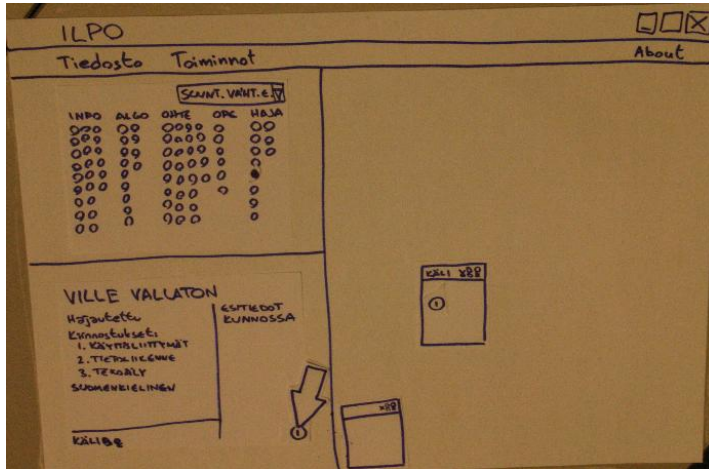
... ja pudottaa pallon ryhmään, jolloin uusi, luotu ryhmä siirtyy keskeemmälle näyttöä. Luodussa ryhmässä näkyy sama Villeä kuvaava pallo. Reunaan jää uusi, tyhjä ryhmä odottamaan, että käyttäjä haluaa luoda uuden ryhmän. Villeä kuvaava pallo muuttuu kaikkialla kuvaamaan, että Ville on sijoitettu yhteen ryhmään.



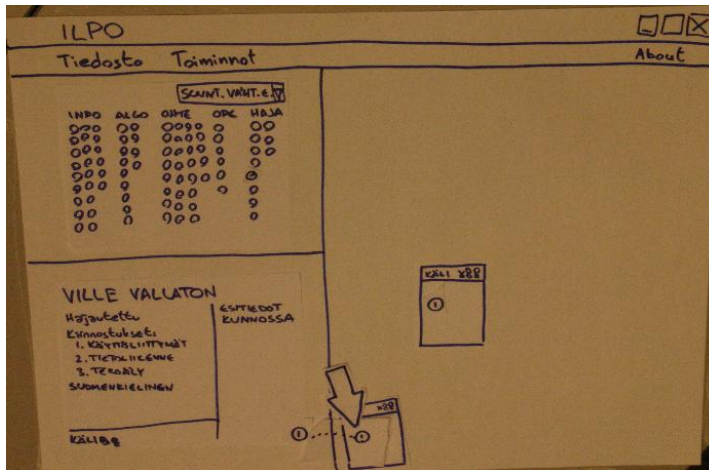
Käyttäjä haluaa nimetä luodun ryhmän ja klikkaa luodun ryhmän yläpalkkia.



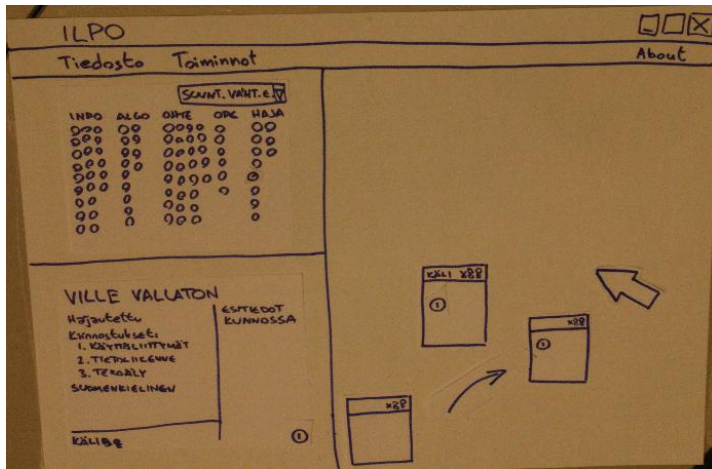
Käyttäjä kirjoittaa ryhmän nimeksi "Käli", sillä Villen ensimmäinen kiinnostuksen kohde olivat käyttöiltyvät.



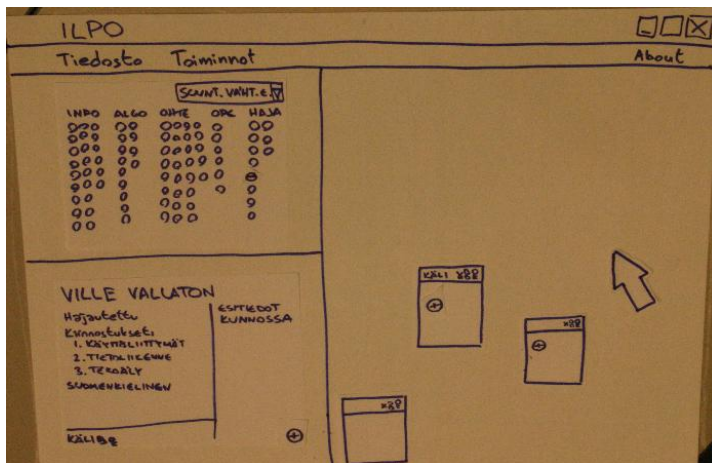
Villen sijoitus "Käli"-ryhmään näkyy Villen tiedoissa. Koska Ville on kiinnostunut myös muista kuin vain käyttöliittymistä, käyttäjä haluaa siirtää Villen myös toiseen ryhmään. Hän ottaa taas kiinni Villen pallosta...



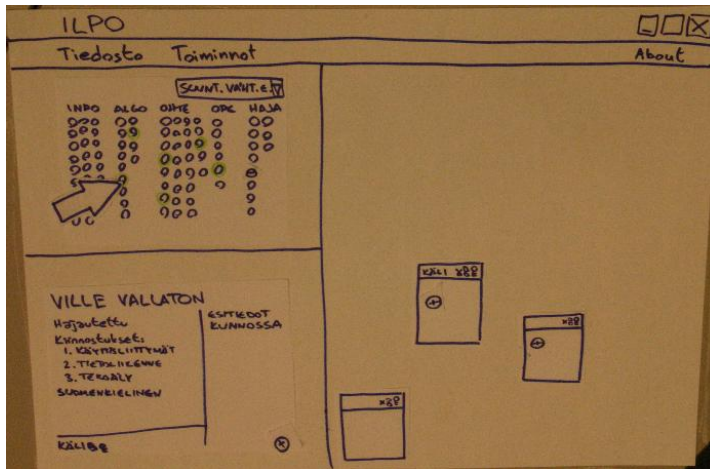
... raahaa pallon uuteen ryhmään ...



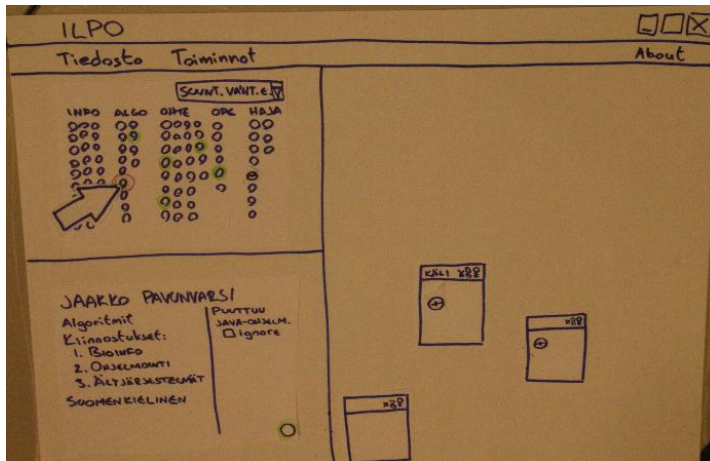
... ja päästää irti, jolloin taas uusi, luotu ryhmä siirtyy keskeemmälle.



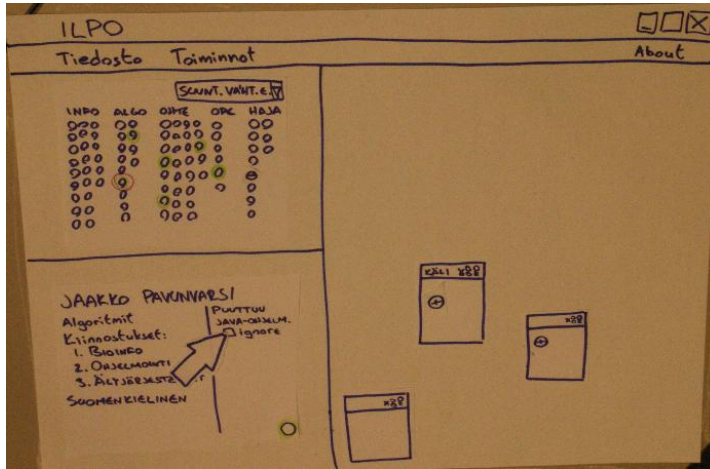
Kaikista Villeä kuvaavista palloista näkee nyt, että Ville on sijoitettu useampaan kuin yhteen ryhmään.



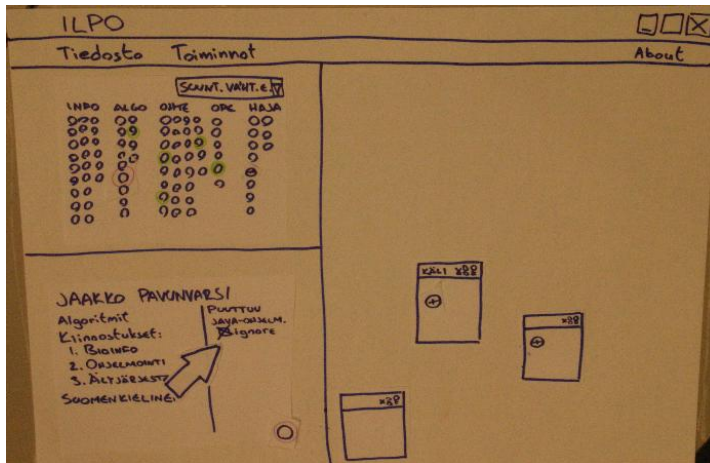
Kaikkien niiden opiskelijoiden pallot, joilla on esitiedoissa puutteita, on aina merkitty tietyllä tavalla (vihreäksi). Käyttäjä haluaa tutustua tällaiseen tapaukseen klikkaamalla tällaista palloa.



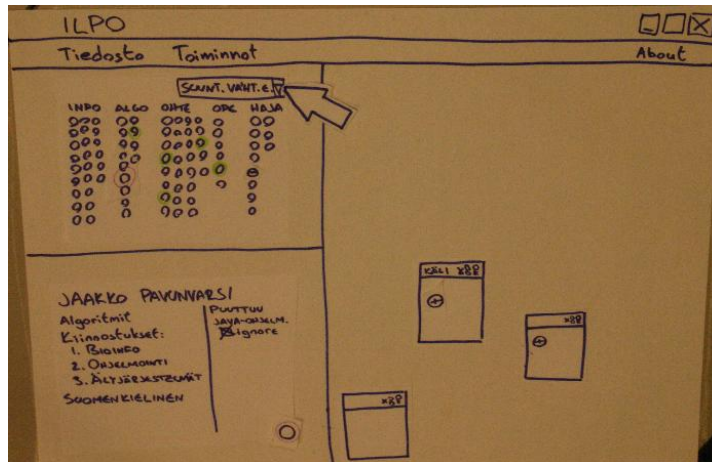
Valittu opiskelija erottuu joukosta erilaisella värityksellä (punainen). Valitun opiskelijan (Jaakko Pavunvarsi) tiedot tulevat alaruutuun. Häneltä näyttäisi puuttuvan Java-ohjelmoinnin suoritus opintorekisteristä.



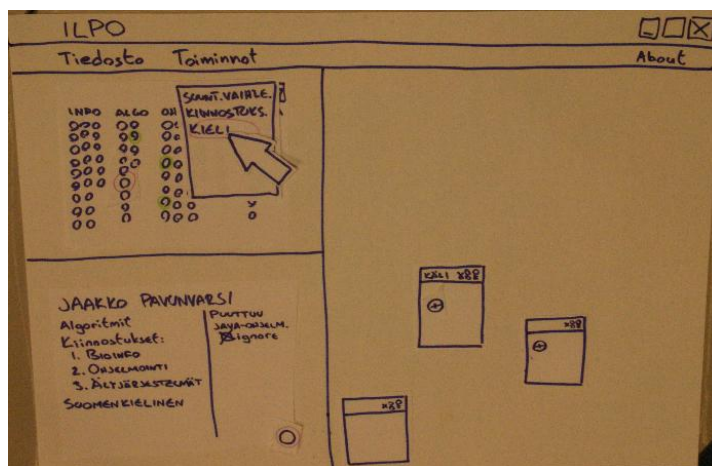
Käyttäjä tietää jotain kautta (ilpon ulkopuolelta), että Jaakko on suorittanut Java-ohjelmoinnin, vaikkei sitä hänen opintorekisterissään näykään. Hän valitsee "ignore" -vaihtoehdon Jaakon Java-ohjelmoinnin puutteen kohdalta.



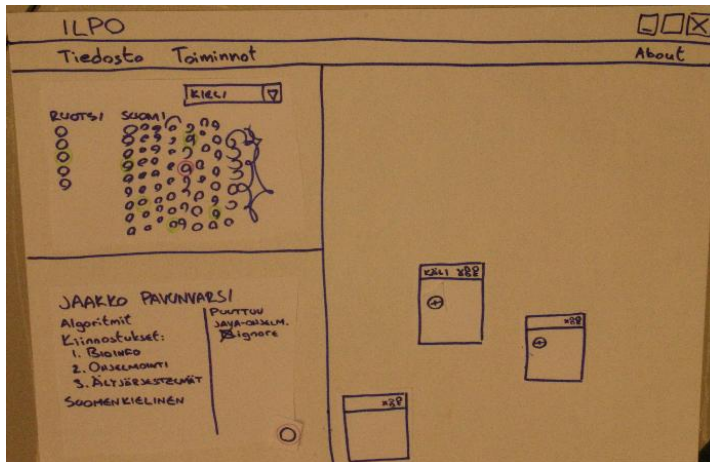
Nyt kun "ignore" on valittu, ei Jaakolla ole enää puutteita esitiedoissa, ja hänen pallonsa väritys muuttuu normaaliksi.



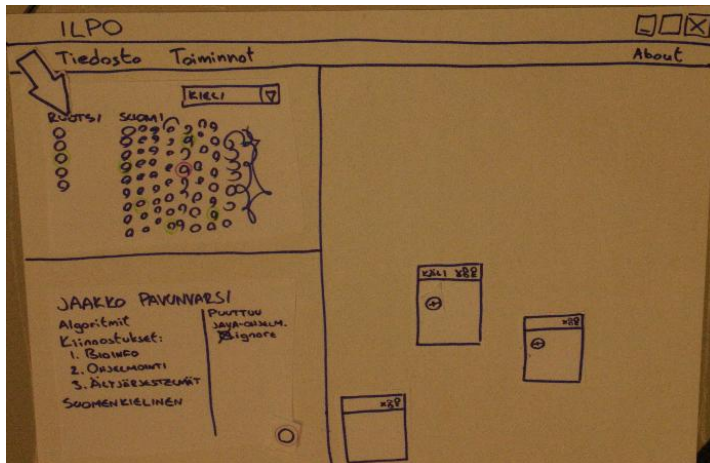
Käyttäjä haluaisi nyt nähdä kaikki ruotsinkieliset opiskelijat. Hän avaa yläruudusta opiskelijoiden järjestämislaitikon ...



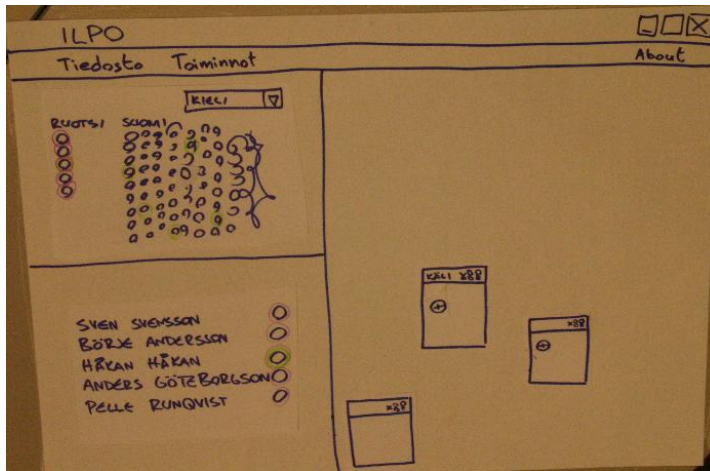
... ja valitsee järjestystavaksi kielen.



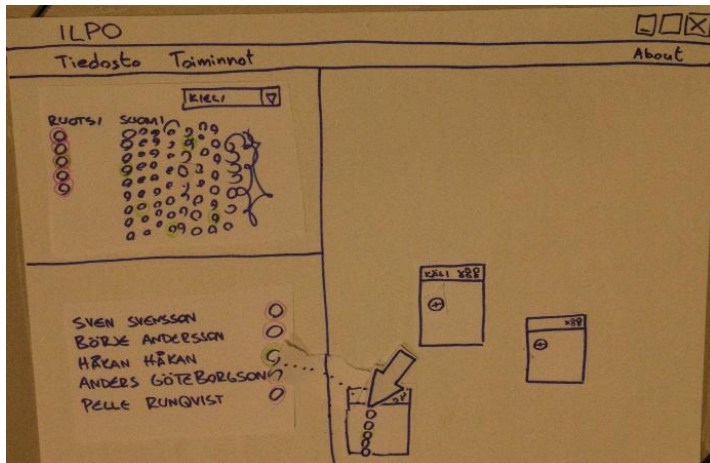
Yläruudun pallot järjestyvät opiskelijoiden kielen mukaan. Ruotsinkielisiä näyttäisi olevan viisi, ja kaikki loput ovat suomenkielisiä.



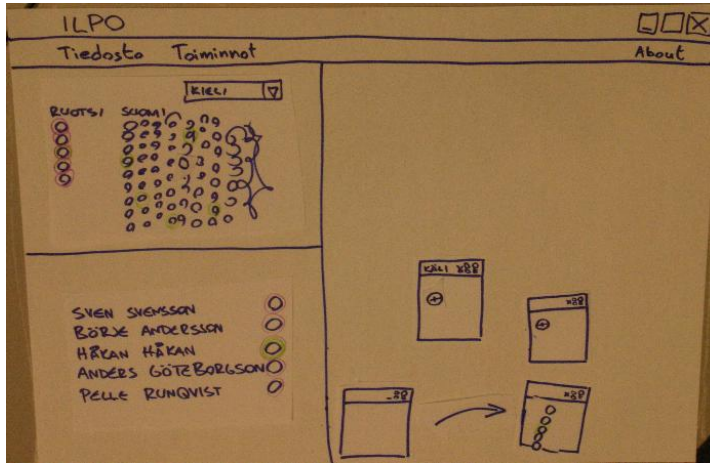
Käyttäjä haluaa muodostaa ruotsinkielisistä oman ryhmän. Niinpä hän valitsee kaikki ruotsinkieliset opiskelijat klikkaamalla "ruotsi" -tulosityhmän otsikkoa.



Kaikkien viiden ruotsinkielisen opiskelijan pallot näkyvät nyt valittuina, ja alaruudussa näkyy lista kaikista valituista sekä heidän pallonsa.



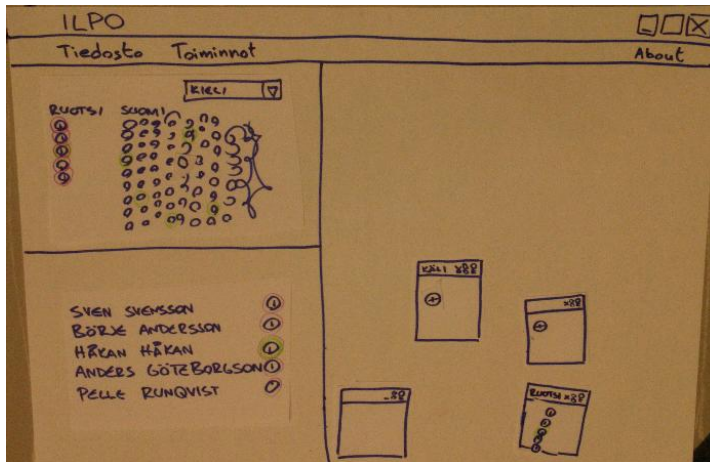
Käyttäjä raahaa kaikki valitut opiskelijat uuteen ryhmään...



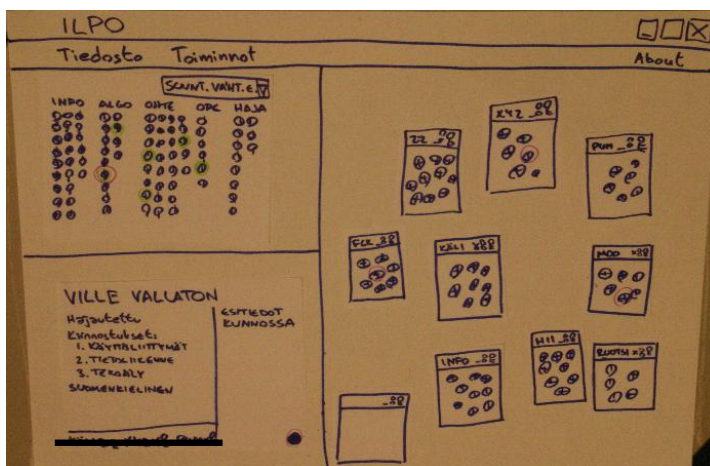
... ja pudottaa heidät sinne, jolloin uusi luotu ryhmä siirtyy keskemmälle.



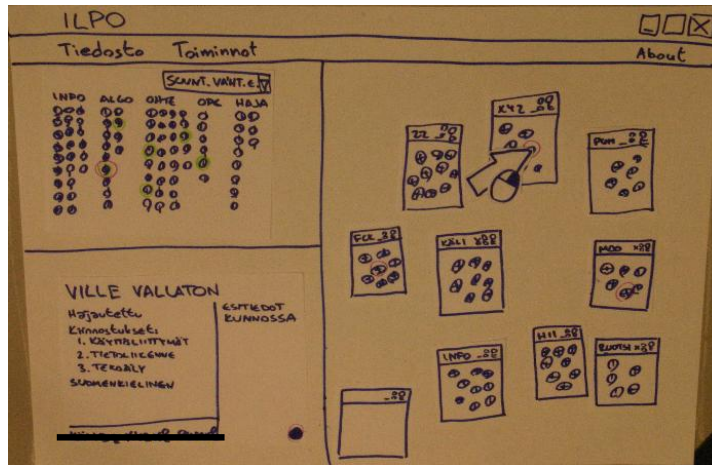
Käyttäjä haluaa nimetä luodun ryhmän. Hän valitsee luodun ryhmän otsikkolaatikon...



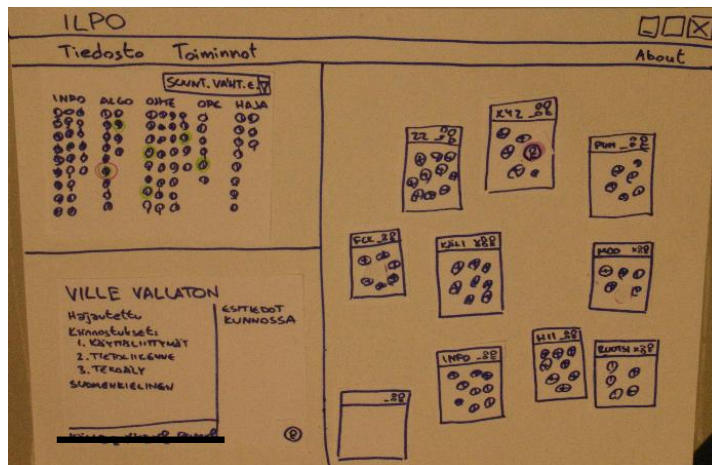
... ja kirjoittaa ryhmän nimeksi "ruotsi".



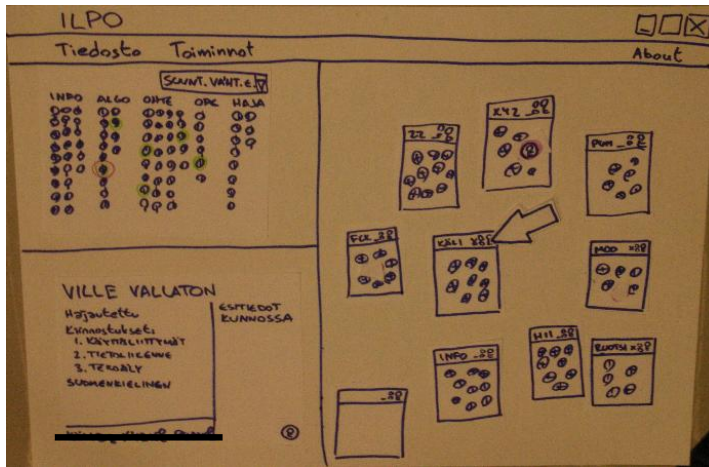
Siirrytään nyt ilpon käytössä eteenpäin. Käyttäjä on sijoittanut jokaisen opiskelijan ainakin yhteen ryhmään, suurimman osan useampaan. Valittu opiskelija Ville Vallaton on sijoitettu kolmeen ryhmään. Tämän näkee sekä alaruudusta että ryhmäruudusta oikealta, kun kaikki Villeä kuvaavat pallot on merkitty punaisella (ryhmissä "XYZ", "FCK" ja "MOO").



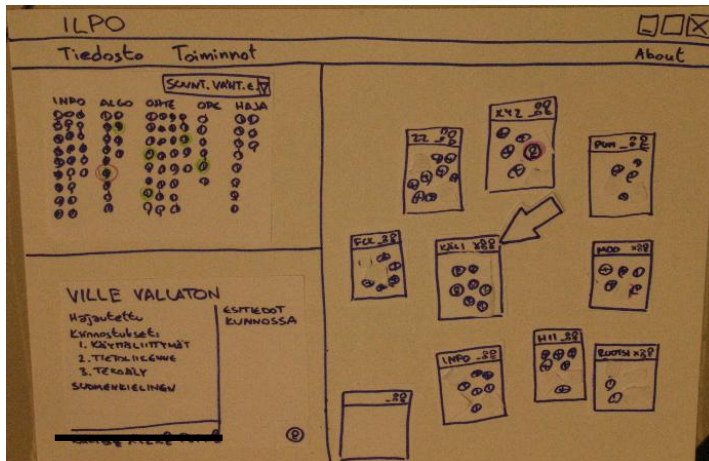
Käyttäjä haluaa lukita Villen "XYZ"-ryhmään. Hän klikkaa oikealla napilla "XYZ" -ryhmässä sijaitsevaa Villen palloa.



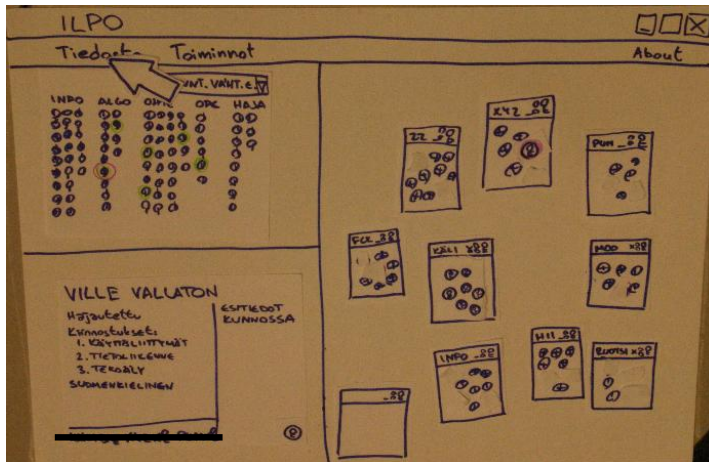
Tällöin Villen pallon symbolin muuttuu kuvaamaan sitä, että ryhmä on lukittu, ja Villeä kuvaavat pallot häviävät kaikista muista kuin valitusta "XYZ"-ryhmästä.



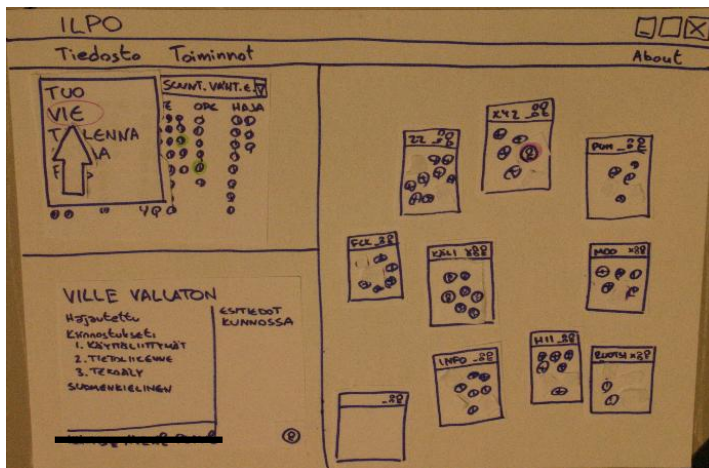
Nyt käyttäjä on tyytyväinen "Käli"-ryhmän kokonaisuuteen. Hän klikkaa ryhmän ylänurkassa olevaa avaimen kuvaa ...



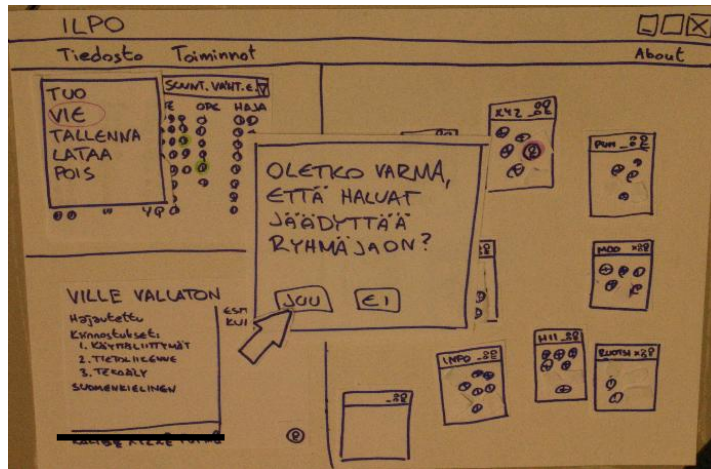
... jolloin kaikki ryhmässä olleet opiskelijat lukittuvat "Käli"-ryhmään ja heitä kuvaavat pallot poistuvat kaikista muista ryhmistä.



Aikansa opiskelijoita siirreltyään ja lukittuaan käyttäjä on mielestään saanut ryhmäjaon valmiiksi.



Käyttäjät valitsee valikosta järjestelyn viemisen ja jäädyttämisen.



Tässä vaiheessa ilpo kysyy käyttäjältä, onko hän aivan varma, että ryhmäjako on hyvä ja että sen voi jäädyttää, sillä tämän jälkeen ryhmäjaolle ei voi tehdä enää mitään ilpolla. Käyttäjä on päätöksestään varma ja valitsee "juu". Kaikki tämän jälkeen tehtävät muutokset ryhmäjakoon tulee tehdä jonkun toisen järjestelmän kautta.

Semmoinen se on se ilpo. Kaipa siihen tulee muitakin yksityiskohtia ja juttuja, mutta tuossa ne olivat ne olennaisimmat. Kiitos mielenkiinnosta ja hauskaa illanjatkoa.