

Kepler-sovellusprojekti

**Joonas Konki
Anu Koskela
Mikko Kuhno
Henrik Paananen
Atte Rätty**

Sovellusraportti

Julkinen
Versio 0.3.0
21.6.2015

**Jyväskylän yliopisto
Tietotekniikan laitos
Jyväskylä**

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2015		
Tilaaja	__.__.2015		
Ohjaaja	__.__.2015		

Tietoa dokumentista

Tekijät:

- Joonas Konki (JK) `joonas.konki@jyu.fi`
- Anu Koskela (AK) `anu.k.koskela@student.jyu.fi`
- Mikko Kuhno (MK) `mikko.kuhno@gmail.com`
- Henrik Paananen (HP) `henrik.j.paananen@student.jyu.fi`
- Atte Rätty (AR) `atte.t.raty@student.jyu.fi`

Dokumentin nimi: Kepler-projekti, Sovellusraportti

Sivumäärä: 73

Tiivistelmä: Kepler-projekti kehitti keväällä 2015 Jyväskylän yliopiston fysiikan laitokselle WWW-sovelluksen oppilaslaboratorion töiden mittausvuorojen varaamiseen. Sovellusraportissa kuvataan projektissa kehitetyn tietojärjestelmän käyttöliittymää, sovelluksen kokonaisrakennetta, tavoitteiden toteutumista, tietokannan rakennetta sekä puutteellisia ja heikkoja toteutusratkaisuja.

Avainsanat: Jatkokehitys, kokonaisrakenne, komponentit, käyttöliittymä, käytänteet, OAuth, palvelin, Pyramid, rajapinnat, REST, tavoitteet, tietojärjestelmä, tietokanta, ylläpito.

Muutoshistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.0.1	10.6.2015	Dokumentin laatiminen aloitettiin.	AR
0.0.2	13.6.2015	Käyttöliittymä näkymien kuvailua.	AR
0.1.0	14.6.2015	Dokumentin puuttuvien lukujen ja alalukujen tekoa.	AR
0.1.1	15.6.2015	Dokumentin puuttuvien osien täydentämistä ja kieliasun parantelua.	AR
0.1.2	16.6.2015	Dokumentin puuttuvien osien täydentämistä, kieliasun parantelua ja ohjaajan palautteen mukaisten korjausten tekemistä.	AR
0.2.0	17.6.2015	Ohjaajan esittämien korjausten tekoa.	AR
0.2.1	18.6.2015	Lukujen 5 ja 6 työstämistä	AR
0.2.2	19.6.2015	Kieliasun parantelua & muotoilua	AR
0.2.3	20.6.2015	Kieliasun parantelua & muotoilua	AR
0.2.4	20.6.2015	Kieliasun parantelua & ohjaajan esittämien muutosten tekoa	AR
0.3.0	21.6.2015	Ohjaajan esittämien muutosten tekoa	AR

Tietoa projektista

Kepler-projekti kehitti keväällä 2015 Jyväskylän yliopiston fysiikan laitokselle WWW-sovelluksen oppilaslaboratorion töiden mittausvuorojen varaamiseen.

Tekijät:

- Joonas Konki (JK) `joonas.konki@jyu.fi`
- Anu Koskela (AK) `anu.k.koskela@student.jyu.fi`
- Mikko Kuhno (MK) `mikko.kuhno@gmail.com`
- Henrik Paananen (HP) `henrik.j.paananen@student.jyu.fi`
- Atte Rätty (AR) `atte.t.raty@student.jyu.fi`

Tilaaaja:

- Sakari Juutinen `sakari.juutinen@phys.jyu.fi`
- Panu Rahkila `panu.rahkila@jyu.fi`

Ohjaajat:

- Petri Partanen `petri.m.partanen@student.jyu.fi`
- Jukka-Pekka Santanen `santanen@mit.jyu.fi`

Yhteystiedot:

- Sähköpostilistat: `keplerit@korppi.jyu.fi` ja `keplerit_opetus@korppi.jyu.fi`
- Sähköpostiarkistot: `http://korppi.jyu.fi/kotka/servlet/list-archive/keplerit/` ja `http://korppi.jyu.fi/kotka/servlet/list-archive/kepler_opetus/`

Sisältö

1	Johdanto	1
2	Termit	2
2.1	Aihealueen termejä	2
2.2	Ohjelmistoja ja teknisiä termejä	3
3	Käyttöliittymä	5
3.1	Sovelluksen näkymät	5
3.2	Ylänavigointialue	7
3.2.1	Käyttäjä-valikko	7
3.3	Navigointipalkki	10
3.4	Auki oleva näkymä	10
3.5	Opiskelijan näkymät	10
3.5.1	Ilmoitustaulu	10
3.5.2	Uusi varaus	11
3.5.3	Omat varaukset	14
3.5.4	Omat ryhmät	16
3.6	Ohjaajan näkymät	18
3.7	Pääkäyttäjän näkymät	19
3.7.1	Hallitse vuoroja	20
3.7.2	Kurssit	22
3.7.3	Työt	24
3.7.4	Resurssit	27
3.7.5	Käyttäjäroolit	28
3.7.6	Muokkaa ilmoitustaulua	29
3.8	Kirjautumissivu	30
4	Sovelluksen kokonaisrakenne ja rajapinnat	35
4.1	Sovelluksen kokonaisrakenne	35
4.2	Ulkopuoliset komponentit	36
4.3	Olenneimmat muutokset toteutusratkaisussa	38
4.4	Muutettavat asetukset	38
4.5	Ulkoiset rajapinnat	42
4.5.1	Korppi-kalenteri	42
4.5.2	OAuth	43

5	Tietokannan rakenne	44
5.1	Kepler-järjestelmän tietokannasta	44
5.2	Permission	45
5.3	RolePermission	45
5.4	Role	45
5.5	UserRole	46
5.6	User	46
5.7	UserInfo	47
5.8	UserInfoType	47
5.9	Language	47
5.10	Translation	48
5.11	UserGroup	48
5.12	UserGroupMember	48
5.13	UserGroupType	49
5.14	TimeSlot	49
5.15	TimeSlotSupervisor	49
5.16	TimeSlotStatus	50
5.17	TimeSlotStatusType	50
5.18	Reservation	50
5.19	ReservationStatus	51
5.20	ReservationStatusType	51
5.21	ReservationUserStatus	51
5.22	ReservationUserStatusType	52
5.23	ReservationNote	52
5.24	Note	52
5.25	NoticeboardNote	53
5.26	Unit	53
5.27	UnitGroupMap	53
5.28	UnitGroup	54
5.29	UnitType	54
5.30	UnitStatus	54
5.31	UnitStatusType	55
5.32	UnitResource	55
5.33	Resource	55
5.34	ResourceStatus	56
5.35	ResourceStatusType	56

6	Tavoitteiden toteutuminen	57
6.1	Vaatimusten toteutuminen	57
6.2	Suoritetut testaukset ja niiden tulokset	58
6.3	Heikot ja puutteelliset toteutusratkaisut	59
6.4	Sovelluksen yleiset jatkokehitysideat	61
6.5	Opiskelijan näkymiin liittyvät jatkokehitysideat	62
6.6	Ohjaajan näkymiin liittyvät jatkokehitysideat	63
6.7	Pääkäyttäjän näkymiin liittyvät jatkokehitysideat	63
6.8	Tietojärjestelmän yleiskäyttöisyys	64
7	Ohjeita ylläpitäjälle ja jatkokehittäjälle	66
7.1	Ohjeita jatkokehittäjälle	66
7.2	Ohjeita ylläpitäjälle	68
8	Yhteenveto	70
	Lähteet	71

1 Johdanto

Jyväskylän yliopiston fysiikan laitoksen oppilaslaboratoriossa opiskelijat voivat suorittaa kokeelliset mittaukset fysiikan perus- ja aineopintotasoihin kursseihin liittyviin laboratoriotöihin. Fysiikan laboratoriotyöt ovat tärkeä osa fysiikan kurssien sisältöä, missä luennoilla opetettua teoriaa pääsee soveltamaan käytäntöön. Kepler-projekti kehitti WWW-sovelluksen fysiikan laitoksen oppilaslaboratorion laboratoriotöiden varausten hallintaan. Varausjärjestelmää tulevat käyttämään pääasiassa fysiikan perus- ja aineopintokurssien opiskelijat, laboratoriovuoroja ohjaavat assistentit sekä varausjärjestelmän pääkäyttäjät. Kehitettävä sovellus korvaa nykyisen WWW-selaimella käytettävän Brahe2007-varausjärjestelmän.

Sovellusprojektissa kehitetyn sovelluksen käyttöliittymän ulkoasu päivitettiin nykyaikaisemmaksi. Vanhasta varausjärjestelmästä kokonaan puuttuva pääkäyttäjän näkymä ja osa sen toiminnoista toteutettiin uudessa järjestelmässä. Lisäksi projektin alussa kartoitettiin muiden yliopiston laitosten varausjärjestelmätarpeet, jotka huomioitiin uuden järjestelmän kehittämisessä ja suunnittelussa.

Sovellusraportin laatimisessa on hyödynnetty Liikkuva-projektin sovellusraporttia [1] sekä Kepler-projektin projektiraporttia [7], projektisuunnitelmaa [8] ja vaatimusmäärittelyä [9]. Muita Kepler-projektin dokumentteja ovat käyttöliittymän [5] ja palvelimen luokkadokumentaatio [6], järjestelmätestausraportti [4], käytettävyydestaussuunnitelma [?] ja käytettävyydestauksesta tehty muistio [?].

Sovellusraportti muodostuu kahdeksasta luvusta. Luvussa 2 kuvataan olennaisia termejä. Luvussa 3 kuvataan sovelluksen käyttöliittymää näkymittäin. Luvussa 4 kuvataan sovelluksen kokonaisrakennetta ja rajapintoja. Luvussa 5 kuvataan tietokannan rakennetta. Luvussa 6 kuvataan heikkoja ja puutteellisia toteutusratkaisuja sekä jatkokehitysideoita. Luvussa 7 annetaan ohjeita ylläpitäjälle ja jatkokehittäjälle.

2 Termit

Luvussa kuvataan projektissa käytettäviä aihealueen, tietojärjestelmän ja toteutus-tekniikoiden termejä.

2.1 Aihealueen termejä

Projektin aihealueen termejä ovat seuraavat:

Assistentti	on fysiikan laboratoriotöiden ohjaajasta käytetty nimitys. Hän on perus-, aine- tai syventäviin opintoihin liittyvien kurssien laboratoriotöitä laboratoriovuoron aikana ohjaava henkilö.
Laboratoriotyö	on mittausvuorolla suoritettava johonkin kurssiin liittyvä työ.
Laboratoriovuoro	on yleensä neljän tunnin mittainen assistentin ohjaama tilaisuus, jossa laboratoriotöihin liittyvät mittaukset voidaan suorittaa.
Mittauslaitteisto	on yhdestä tai useammasta mittalaitteesta koostuva laitteisto, jolla laboratoriotyön mittaukset suoritetaan.
Mittausryhmä	on opiskelijajoukko, joka on varannut ajan laboratoriotyön mittaamiseen jollekin ohjatulle laboratoriovuorolle.
Mittausvuoro	on opiskelijan joltakin laboratoriovuorolta varaama aika jonkin tietyn laboratoriotyön mittaamiseen.
Ohjaaja	on assistentin käyttäjäroolin nimi kehitetyssä varausjärjestelmässä.
Ohjausvuoro	on ohjaajan laboratoriovuoro, jolle hänet on asetettu ohjaajaksi.
Opiskelija	on mittausvuoroja varaavan opiskelijan käyttäjäroolin nimi kehitetyssä varausjärjestelmässä.

Oppilaslaboratorio	on fysiikan laitoksen oppilaslaboratorio, jonka vuorojen varaukseen järjestelmä kehitettiin.
Pääkäyttäjä	on oppilaslaboratorion johtajan käyttäjäroolin nimi kehitetyssä varausjärjestelmässä. Pääkäyttäjä on henkilö, joka hallitsee toisten käyttäjien rooleja ja lisää laboratoriotöitä järjestelmään.

2.2 Ohjelmistoja ja teknisiä termejä

Dokumentissa käytettyjä teknisiä termejä ovat seuraavat:

Back end	on ohjelmiston se osa, joka on kauimpana ohjelmiston käyttäjästä. WWW-sovelluksessa se on palvelinpuolen ohjelmisto.
Bootstrap	on WWW-dokumenttien ulkoasun muokkausta helpottava, HTML-kuvauskieltä ja CSS-tyylikieltä käyttävä kirjasto.
CSRF	on tietoturvahyökkäys, joka pakottaa käyttäjän suorittamaan hänen haluamattaan toimintoja WWW-sovelluksessa, johon hän on kirjautunut. Termi tulee sanoista <i>cross-site request forgery</i> .
CSS	on tyylikieli WWW-dokumenttien ulkoasun määrittämiseen.
Front end	on ohjelmiston se osa, joka on lähimpänä ohjelmiston käyttäjää. Käytännössä front end sisältää sovelluksen käyttöliittymän ja sen toimintalogiikan.
HTML5	on uusin versio WWW-dokumenttien HTML-kuvauskielestä.
HTTPS	on WWW:ssä suojattuun tietoliikenteeseen käytetty protokolla.
iCalendar	on kalenteritiedostomuoto, jolla kalenterisovellukset pystyvät vaihtamaan kalenteritietoja keskenään.

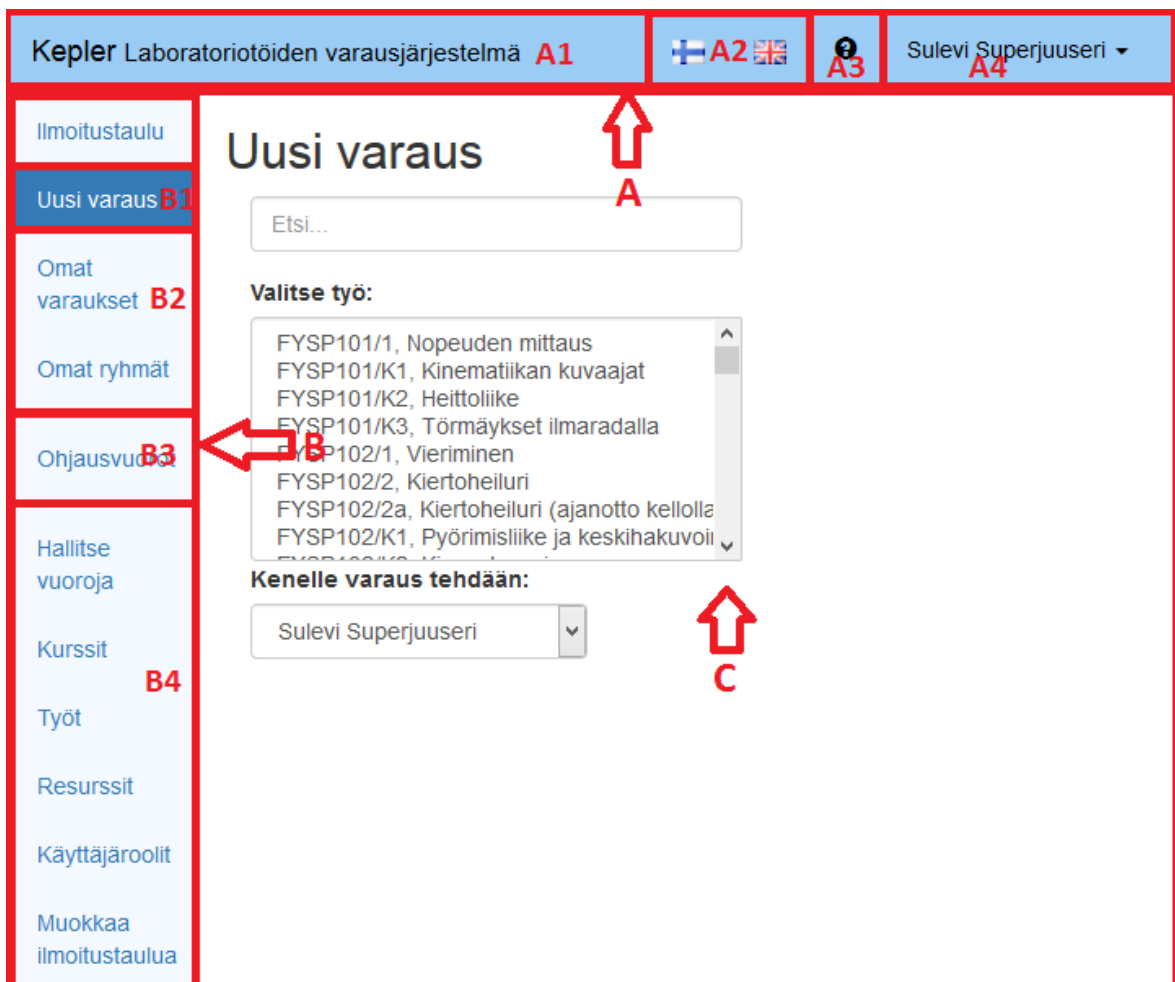
JavaScript	on WWW-sovelluksissa käytetty ja WWW-selaimissa toimiva dynaamisesti tyypitetty ohjelmointikieli.
jQuery	on WWW-dokumenttien dynaamisen sisällön käsittelyä helpottava avoimen lähdekoodin lisenssin alainen JavaScript-kirjasto.
JYU-tunnukset	ovat Jyväskylän yliopiston IT-palveluiden käyttäjän henkilökohtaiset käyttäjätunnukset.
Korppi	on Jyväskylän yliopiston opintotietojärjestelmä.
Käyttöliittymä	on ohjelmiston osa, jonka kautta käyttäjä käyttää ohjelmistoa.
OAuth2	on sovelluksessa autentikointiin käytettävä protokolla.
ORM	on tekniikka, jolla tietokannan tietoja pystytään helposti käsittelemään suoraan ohjelmointikielen olioiden kautta.
Pyramid	on Python-ohjelmointikielillä toteutettu minimalistinen WWW-sovelluskehys.
Python	on dynaamisesti tyypitetty korkeantason ohjelmointikieli.
Sovelluskehys	on ohjelman rungon muodostava teknologia.
SQL	on yleisesti käytetty relaatiotietokantastandardi.
SQLAlchemy	on avoimen lähdekoodin SQL- ja ORM-kirjasto Python-kielille.
Tietokanta	on tietovarasto, joka tyypillisesti mallintaa jollain tapaa tiedon yhteyksiä toisiin tietoihin. Tällöin puhutaan relaatiotietokannasta.
WWW-sovellus	on WWW-tekniikoita hyödyntävä ja selaimella käytettävä sovellus.
WWW-sovelluskehys	on sovelluskehys, joka on suunniteltu nopeuttamaan WWW-sovellusten kehittämistä.

3 Käyttöliittymä

Luvussa esitellään sovelluksen käyttöliittymän eri näkymät sekä niihin liittyvät toiminnot ja niiden väliset suhteet.

3.1 Sovelluksen näkymät

Kuvassa 3.1 havainnollistetaan sovelluksen yleistä näkymää, kun on kirjautuneena sisään pääkäyttäjänä.



Kuva 3.1: Keplerin yleisnäkymä pääkäyttäjänä.

Sovelluksen käyttöliittymä koostuu seuraavista osista:

- ylänavigointialue (osoitettu kirjaimella A),
- järjestelmän nimi ja kuvaus (A1),
- kielen valinta (A2),
- ohjeisiin vievä painike (A3),
- käyttäjä-valikko (A4),
- navigointipalkki (B),
- valitun näkymän korostus (B1),
- opiskelijan näkymiä (B2, sisältää myös tällä hetkellä korostetun Ilmoitustaulu-näkymän),
- ohjaajan näkymä (B3),
- pääkäyttäjän näkymiä (B4),
- auki oleva näkymä (C)

3.2 Ylänavigointialue

Kepler-järjestelmän kaikissa näkymissä (kirjautumissivua lukuun ottamatta) on nähtävissä ylänavigointialue, joka esitetään kuvassa 3.2.



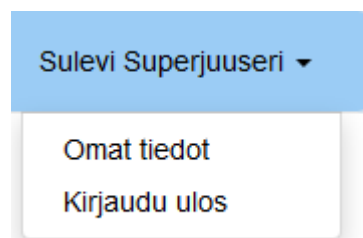
Kuva 3.2: Yläpalkki.

Ylänavigointialue koostuu neljästä osasta:

Järjestelmän nimi ja kuvaus	Järjestelmän nimi ja kuvaus kertovat käyttäjälle hänen käyttämänsä järjestelmän nimen ja lyhyen kuvauksen sen käyttötarkoituksesta.
Kielen valinta	Kielen valinnassa esitetään kielivaihtoehdot lippuina. Lippua painamalla järjestelmän käyttämä kieli vaihtuu tämän istunnon ajaksi. Pysyvän kielivalinnan voi tehdä <i>Omat tiedot</i> -näkyvässä.
Ohjeisiin vievä painike	Järjestelmän ohjeisiin pääsee painamalla kielen valinnan vieressä esitettyä kysymysmerkkipainiketta. Ohjenäkymä on koko järjestelmälle yleinen (se ei siis muutu sivukohtaisesti).
Käyttäjä-valikko	Käyttäjävalikkoa kuvaillaan alaluvussa 3.2.1.

3.2.1 Käyttäjä-valikko

Käyttäjä voi omaa nimeään painamalla avata valikon, josta paljastuu kaksi uutta toimintoa: *Omat tiedot* ja *Kirjaudu ulos*.



Kuva 3.3: Käyttäjä-valikko.

Omat tiedot-vaihtoehtoa painamalla aukeaa seuraavanlainen näkymä.

Korpista haetut käyttäjätiedot

Käyttäjätunnus: susupe
Nimi: Sulevi Superjuuseri
Sähköposti: admin@kepler.jyu.fi

Muut käyttäjätiedot

Käyttäjäroolisi: Pääkäyttäjä
Opiskelija
Ohjaaja, Perusopinnot
Ohjaaja, Aineopinnot

Oletuskieli: Suomi
 English

Tallenna muutokset

Kuva 3.4: Omat tiedot.

Kuvan 3.4 kuvaamassa näkymässä käyttäjä näkee ne tietonsa, jotka tietojärjestelmästä löytyvät. Käyttäjä voi vaihtaa oletuskieltään valitsemalla haluamansa kielen ja painamalla *Tallenna muutokset*-painiketta.

Kirjaudu ulos-valintaa painamalla käyttäjä voi kirjautua ulos tietojärjestelmästä.

3.3 Navigointipalkki

Navigointipalkkia (kirjain B kuvassa 3.1) käytetään eri näkymien välillä siirtymiseen. Sen sisältö muuttuu sisäänkirjautuneen käyttäjän roolien mukaan. Kuvassa 3.1 sisäänkirjautuneella käyttäjällä on opiskelijan, ohjaajan ja pääkäyttäjän roolit, joten hän näkee navigointipalkin kaikki vaihtoehdot.

Navigointipalkissa (ja ylänavigointipalkissa) on auki oleva näkymä korostettuna tummemmalla taustavärillä.

3.4 Auki oleva näkymä

Auki olevassa näkymässä (kirjain C kuvassa 3.1) esitetään kulloinkin auki olevan näkymän toiminnallisuus. Suurin osa tietojärjestelmän toiminnallisuuksista suoritetaan käyttäen kulloinkin valittuna olevaa näkymää.

3.5 Opiskelijan näkymät

Opiskelijan roolin omaavalla käyttäjällä on oletuksena käytössään seuraavat näkymät: *Ilmoitustaulu*, *Uusi varaus*, *Omat varaukset* ja *Omat ryhmät*.

3.5.1 Ilmoitustaulu

Ilmoitustaululla käyttäjät näkevät pääkäyttäjän järjestelmään lisäämiä ilmoituksia. Ominaisuus havaittiin hyväksi Kepleriä edeltäneessä Brahe2007-järjestelmässä.

Ilmoitustaulu

Lisää vuoroja tulossa!	Sulevi Superjuuseri	21.6.2015 15.35
Laitoksen johto on luvannut opiskelijalaboratorioille lisää resursseja! Tämä tarkoittaa sitä, että myös uusia vuoroja on tulossa. Oppilaat voivat ilmoittaa ehdotuksia uusien vuorojen ajaksi.		
Muistakaa laskimenne!	Sulevi Superjuuseri	21.6.2015 15.34
Muistakaa ottaa laskimenne mukaan, kun tulette labroihiin! Laitoksella on varalaskimia hyvin rajoitetusti.		

Kuva 3.5: Ilmoitustaulu.

3.5.2 Uusi varaus

Uusi varaus-näkyvässä käyttäjät voivat tehdä uusia varauksia itselleen ja ryhmilleen. Varauksen tehdäkseen käyttäjän pitää valita, mitä työtä varten ja kenelle varaus tehdään.

Uusi varaus

Valitse työ:

- FYSP101/1, Nopeuden mittaus
- FYSP101/K1, Kinematiikan kuvaajat
- FYSP101/K2, Heittoliike
- FYSP101/K3, Törmäykset ilmaradalla
- FYSP102/1, Vieriminen
- FYSP102/2, Kiertoheiluri
- FYSP102/2a, Kiertoheiluri (ajanotto kellolla)
- FYSP102/K1, Pyörimisliike ja keskihakuvoii

Työn nimi

FYSP102/1, Vieriminen

KuvausTyön vastuuhenkilöt ovat A. Assari ja O. Ohjaaja. Työn ohjeet: [www](#)**Mittausryhmän maksimikoko**

2 jäsentä

Kenelle varaus tehdään:

Kuva 3.6: Uuteen varaukseen liittyvien valintaperusteiden valinta.

Varauksen ajankohta voidaan valita käyttäen joko kalenteri- (kuvassa 3.7) tai lista-näkymää (kuvassa ??).

Päivä Viikko Kuukausi Tänään < >

15 — 21.6.2015

	ma - 15.06	ti - 16.06	ke - 17.06	to - 18.06	pe - 19.06
8:00					
9:00					
10:00					
11:00					
12:00					
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					

Anna halutessasi lisätietoja:

Tee varaus

Kuva 3.7: Uuden varauksen ajankohdan valinta kalenterin avulla.

Valitse sopiva mittausvuoro:

Vaihda kalenterinäkymään

Ohjausvuorot:

25.6.2015 8.00 - 12.00, Perusopinnot

29.6.2015 8.00 - 12.00, Perusopinnot

Anna halutessasi lisätietoja:

Tee varaus

Kuva 3.8: Uuden varauksen ajankohdan valinta listan avulla

3.5.3 Omat varaukset

Tässä näkymässä käyttäjä näkee omat ja ryhmiensä varaukset, niihin liittyviä tietoja sekä pystyy perumaan niitä (mikäli kyseinen vuoro on vielä peruttavissa). Käyttäjä voi myös valita tarkasteltavan ajanjakson.

Omat varaukset

Menneet vuorot Menossa nyt

Vaihda kalenterinäkömään

Tulevat vuorot

Näytetään vuorot ajalta:

Kuukausi ▼

➤	24.6.2015 16.00 - 20.00	FYSP110/K2, Oskilloskoopin syventävä käyttö
➤	29.6.2015 8.00 - 12.00	FYSA242/K1, Terminen elektroniemissio Tämä vuoro on peruttu!
▼	30.6.2015 12.00 - 16.00	FYSP102/2, Kiertoheiluri
Ryhmä: Sulevi Superjuuseri		
Peru varaus		

Kuva 3.9: Käyttäjän omat varaukset listamuodossa.

Omat varauksensa on mahdollista nähdä myös kalenterimuodossa.

Päivä Viikko Kuukausi 29.6 — 5.7.2015 Tänään < >

	ma - 29.06	ti - 30.06	ke - 01.07	to - 02.07	pe - 03.07
8:00	08.00 - 12.00 FYSA242/K1 Terminen elektroniemissio Tämä vuoro on peruttu! Henkilökohtainen				
9:00					
10:00					
11:00					
12:00		12.00 - 16.00 FYSP102/2 Kiertoheiluri Henkilökohtainen varaus			
13:00					
14:00					
15:00					
16:00					
17:00					
18:00					
19:00					

Peru valittu varaus

Kuva 3.10: Käyttäjän omat varaukset kalenterimuodossa.

3.5.4 Omat ryhmät

Omat ryhmät-näkyvässä käyttäjä voi tarkastella ryhmiä, joihin hän kuuluu. Hän voi poistaa luomiaan ryhmiä, joiden omistaja hän on, ja hän voi luoda uusia ryhmiä. Kuvassa 3.11 on esitetty käyttäjän ryhmien listaus ja uuden ryhmän luomiseen käytettävä lomake.

Omat ryhmät

▼ Keplot	2
▼ Labraryhmä	2
Sulevi Superjuuseri	
Olli Opiskelija	Ryhmän omistaja
<button>Poista ryhmä</button>	

[Tee uusi ryhmä](#)

Uuden ryhmän luominen

Ryhmän nimi

Hae käyttäjä

Ryhmän jäsenet

- Olli Opiskelija

Tallenna

Peruuta

Kuva 3.11: Omat ryhmät ja uuden ryhmän luominen.

3.6 Ohjaajan näkymät

Ohjaajan roolin omaavalla käyttäjällä on oletuksena käytössään (vain) *Ohjausvuorot*-näkyvä. Huomaa, että ainakin Jyväskylän yliopiston fysiikan laitoksen käyttötapauksessa lähes kukaan ei kuitenkaan ole rooliltaan pelkästään ohjaaja, sillä ohjaajat ovat joko laitoksen opiskelijoita tai henkilökuntaa.

Ohjausvuorot-näkyvässä käyttäjä näkee omat ohjausvuoronsa. Käyttäjä voi valita miltä ajanjaksolta hän haluaa vuoroja tarkastella. Käyttäjä voi myös nähdä listauksen ohjausvuorojensa osallistujista ja heidän yhteystiedoistaan. Käyttäjä voi nähdä ohjausvuoronsa lista- tai kalenterinäkyvässä ja hän voi valita, haluaako nähdä myös muiden ohjaajien ohjausvuorot.

Ohjausvuorot


Menneet vuorot Menossa nyt
 Tulevat vuorot

Näytä omat vuorot
 Näytä kaikkien vuorot

Vaihda kalenterinäkymään

Näytetään vuorot ajalta:

Kuukausi ▼

▼	22.6.2015 8.00 - 12.00	Perusopinnot, Kalle Kalastaja	2
▼	22.6.2015 8.00 - 12.00	Aineopinnot, Kalle Kalastaja	1
FYSA242/K1, Terminen elektroniemissio  Olli Opiskelija			
▼	23.6.2015 12.00 - 16.00	Aineopinnot, Kalle Kalastaja	1
▼	24.6.2015 16.00 - 20.00	Perusopinnot, Kalle Kalastaja	1
▼	26.6.2015 12.00 - 16.00	Aineopinnot, Kalle Kalastaja	1
▼	29.6.2015 8.00 - 12.00	Perusopinnot, Kalle Kalastaja	2
▼	29.6.2015 8.00 - 12.00	Aineopinnot, Kalle Kalastaja Tämä vuoro on peruttu!	0

Kuva 3.12: Ohjausvuorot listana.

3.7 Pääkäyttäjän näkymät

Pääkäyttäjän roolin omaavalla käyttäjällä on oletuksena käytössään seuraavat näkymät: *Hallitse vuoroja, Kurssit, Työt, Resurssit, Käyttäjäroolit ja Muokkaa ilmoitustaulua.*

3.7.1 Hallitse vuoroja

Hallitse vuoroja-näkyvässä käyttäjä näkee kaikki Kepler-järjestelmän vuorot. Hän voi rajata tarkasteltavien vuorojen ajankohdaksi menneet, menossa olevat tai tulevat vuorot. Vuorosta esitetään tietoina mihin opintokokonaisuuteen vuorolla tehtävät laboratoriotyöt kuuluvat, vuoron ohjaajat ja lista vuoron osallistujista yhteystietoineen. Vuoroja pystyy myös perumaan käyttämällä *Peru*-painiketta.

Tarkoituksena oli myös toteuttaa mahdollisuus esittää vuorot kalenterimuodossa ja tarjota mahdollisuus vuorojen muokkaamiseen, mutta nämä ominaisuudet sovittiin tilaajan kanssa jätettävän jatkokehitykseen.

Hallitse ohjausvuoroja

Menneet vuorot Menossa nyt


Vaihda kalenterinäkymään

Tulevat vuorot

Näytetään vuorot ajalta:

Kuukausi ▼

Lisää uusi vuoro...

▼	22.6.2015 8.00 - 12.00	Perusopinnot, Kalle Kalastaja	2
▼	22.6.2015 8.00 - 12.00	Aineopinnot, Kalle Kalastaja	1
FYSA242/K1, Terminen elektroniemissio			
 Olli Opiskelija			
			Muokkaa Poista
▼	23.6.2015 8.00 - 12.00	Perusopinnot, Maria Assari	0
▼	23.6.2015 12.00 - 16.00	Aineopinnot, Kalle Kalastaja	1
▼	23.6.2015 16.00 - 20.00	Perusopinnot, Maria Assari Tämä vuoro on peruttu!	0

Kuva 3.13: Hallitse vuoroja.

[Lisää uusi vuoro...](#)

Uuden vuoron luominen

Päivämäärä:

Alku- ja loppuaika: 8:00 - 12:00
 12:00 - 16:00
 16:00 - 20:00
 Muu: -

Työvarauksia max:

Opintokokonaisuus:

Ohjaaja:

Vuoron status:

Kuva 3.14: Uuden vuoron lisääminen.

3.7.2 Kurssit

Kurssit-näkylässä listataan tietojärjestelmän sisältämät kurssit. Listattujen kurssien tietoja voi muokata käyttämällä oikean reunan syöttökenttiä ja painamalla *Tallenna muutokset*-painiketta. Vanhoja kursseja pystyy poistamaan ja uusia kursseja luomaan.

Hallitse kursseja

Etsi...

FYSA210
FYSA220
FYSA230
FYSA241
FYSA242
FYSP101
FYSP102
FYSP103
FYSP104
FYSP105
FYSP106
FYSP107
FYSP110

Kurssin koodi: FYSP101

Opintokokonaisuus: Perusopinnot

Työt: K2, K1, 1, K3

Heittoliike
Kinematiikan kuvaajat
Nopeuden mittaus
Törmäykset ilmaradalla

+ Lisää työ

Tallenna muutokset Poista kurssi

+ Lisää uusi kurssi

Kuva 3.15: Hallitse kursseja.

Kursseihin liittyvien (laboratorio)työiden tietoja voi muokata *Kurssit*-näkyvässä ja kurssiin voi lisätä siihen liittyviä töitä.

Kurssin koodi: Anna kurssille koodi. Esim. FYSP100.

Opintokokonaisuus: Perusopinnot

Työt: Kurssille ei ole määritetty töitä.

+ Lisää työ

Tallenna muutokset Poista kurssi

Kuva 3.16: Lisää uusi kurssi.

Valitse työ:



- Braggin diffraktio
- Dopplerin ilmiö
- Elektronin diffraktio
- Elektronin liike magneettikentässä
- Franckin ja Hertzin koe
- Fraunhoferin diffraktio**
- Geigerin ja Müllerin putki
- Germaniumin energia-aukko
- Hallin ilmiö
- Heittoliike
- Helmholzin kelat
- Hitausmomentti
- Höyrynpaine ja höyrystymislämpö
- Kaasulämpömittari
- Kaasututkimus
- Kiintoliike

Valitse työ:

Nimi

Fraunhoferin diffraktio

KuvausTyön vastuuhenkilöt ovat A. Assari ja O. Ohjaaja. Työn ohjeet: [www](#)**Mittausryhmän maksimikoko**

2

Anna koodi tälle työlle:

Kuva 3.17: Työn lisääminen kurssiin.

3.7.3 Työt

Työt-näkyessä listataan tietojärjestelmän sisältämät työt. Listattujen töiden tietoja voi muokata käyttämällä oikeassa reunassa olevia syöttökenttiä ja painamalla *Tallenna muutokset*-painiketta. Vanhoja töitä pystyy poistamaan ja uusia töitä luomaan.

Laboratoriotyöt

Etsi...

Braggin diffraktio
Dopplerin ilmiö
Elektronin diffraktio
Elektronin liike magne
Franckin ja Hertzin ko
Fraunhoferin diffraktio
Geigerin ja Müllerin p
Germaniumin energia
Hallin ilmiö
Heittoliike
Helmholzin kelat
Hitausmomentti
Höyrynpaine ja höyry
Kaasulämpömittari
Kaasututkimus

+ Lisää uusi työ

Nimi (suomi): Dopplerin ilmiö

Nimi (English): Doppler effect

Kuvaus (suomi): Työn vastuuhenkilöt ovat A. Assari ja O. Ohjaaja. Työn ohjeet: www

Kuvaus (English): The supervisors in charge of the experiment are A. Assari and O. Ohjaaja. Experiment instructions: www

Osallittujia max: 2

Resurssit: 3 **Ge-ilmais**
2 **Pyöriväalusta**

+ Lisää resurssi

Tallenna muutokset

Poista työ

Kuva 3.18: Hallitse töitä.

Nimi
(suomi):

Anna työlle nimi.

Nimi
(English):

Give a name to the experiment.

Kuvaus
(suomi):

Anna tarkempia tietoja työstä.

Kuvaus
(English):

Give more detailed information about the experiment.

Osallittujia
max

1

Resurssit:

Työlle ei ole määritelty resursseja

+ Lisää resurssi

Tallenna muutokset

Poista työ

Työt vaativat käyttöönsä tietyn määrän resursseja. *Kurssit*-näkyvässä tämän määrän voi määrittellä ja töihin voi lisätä uusia resursseja. Työhön liittyviä resursseja voi poistaa muuttamalla työn kyseistä resurssia vaatiman määrän arvoksi nolla ja tallentamalla muutokset.

Valitse resurssi: ✕

Datastudio

Ilmarata

Nimi
Ilmarata

Kuvaus
Ei kuvausta

Käytettävissä Yhteensä
3

Montako tässä työssä käytetään:

1 ▲▼

Lisää Peruuta

Kuva 3.20: Resurssi lisääminen työhön.

3.7.4 Resurssit

Resurssit-näkyvässä listataan tietojärjestelmästä löytyvät resurssit. Resurssien tietoja voi muokata oikealle olevista syöttökentistä. Muokatut tiedot tallennetaan käyttämällä *Tallenna muutokset*-painiketta. Lisäksi resursseista on olemassa tieto siitä, kuinka monta resurssia on missäkin tilassa. Myös kokonaan uusia resursseja voi lisätä. Resursseja ei voi poistaa järjestelmästä, mutta ne voi kirjata sellaisiin tiloihin, jossa niitä ei voi käyttää tai niiden määrän voi muuttaa nolleen.

Resurssit

- Datastudio**
- Ge-ilmainen
- Ilmarata
- Pyöriväalusta

Nimi (suomi):

Nimi (English):

Kuvaus (suomi):

Kuvaus (English):

Käytettävissä:

Pois käytöstä:

Rikki: **Yhteensä**

Kuva 3.21: Hallitse resursseja.

3.7.5 Käyttäjäroolit

Käyttäjäroolien hallintaan ei nykyisellään ole toimivaa käyttöliittymää. Käyttäjärooleihin liittyvän toiminnallisuuden yhdistäminen käyttöliittymään sovittiin tilaa-
jan kanssa jätettävän jatkokehitykseen. Nykyinen *Käyttäjäroolit*-näkyminen on sekava ja käytettävyydeltään heikko. Koska se olisi syytä tehdä kokonaan uusiksi ennen käyttöönottoa, sitä ei esitetä.

3.7.6 Muokkaa ilmoitustaulua

Muokkaa ilmoitustaulua-näkylässä käyttäjä näkee ilmoitustaululla esitettävät ilmoitukset. Hän voi myös muokata niitä tai luoda kokonaan uusia ilmoituksia.

Muokkaa ilmoitustaulua

[Lisää viesti](#) Näytä piilotetut

Lisää vuoroja tulossa! Sulevi Superjuuseri 21.6.2015 15.35

Laitoksen johto on luvannut opiskelijalaboratorioille lisää resursseja! Tämä tarkoittaa sitä, että myös uusia vuoroja on tulossa. Oppilaat voivat ilmoittaa ehdotuksia uusien vuorojen ajaksi.

[Muokkaa](#) [Piilota](#)

Muistakaa laskimenne! Sulevi Superjuuseri 21.6.2015 15.34

Muistakaa ottaa laskimenne mukaan, kun tulette labroihiin! Laitoksella on varalaskimia hyvin rajoitetusti.

[Muokkaa](#) [Piilota](#)

Kuva 3.22: Ilmoitusten hallinnointi.

Muokkaa ilmoitusta ×

Oppilaat huomio!

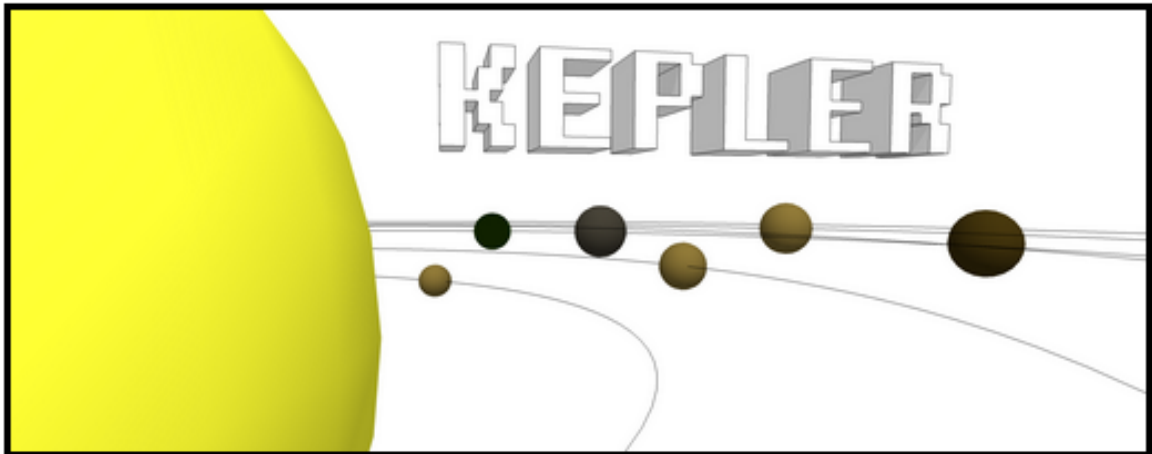
Nyt testataan Kepleriä!

Tallenna

Kuva 3.23: Lisää uusi ilmoitus.

3.8 Kirjautumissivu

Käyttäjän syöttäessä selaimensa Kepler-järjestelmän osoitteen, hän saapuu kuvan 3.24 kirjautumissivulle. Tältä sivulta hän voi kirjautua järjestelmään sisään.



Kepler

Laboratoriotöiden varausjärjestelmä

[Log in](#)

Kuva 3.24: Kirjautumissivu.

Tältä sivulta käyttäjä ohjataan Jyväskylän yliopiston OAuth-palveluun.



Kirjaudu sisään

[fi](#) [en](#)

Käyttäjätunnus

Salasana

[Kirjaudu >](#)

Kuva 3.25: Jyväskylän yliopiston OAuth-sivu.

Kirjautumisen jälkeen OAuth-palvelu pyytää lupaa antaa Kepler-järjestelmän käyttää joitain tiettyjä palveluita. Ilman näitä lupia, Kepler ei toimi.



Lupapyyntö

fi

en

Palvelu Kepler-sovelluksen paikallinen palvelin haluaa lupia seuraaviin oikeuksiin, sallitaanko?

call_current_user_info

Kyllä

En

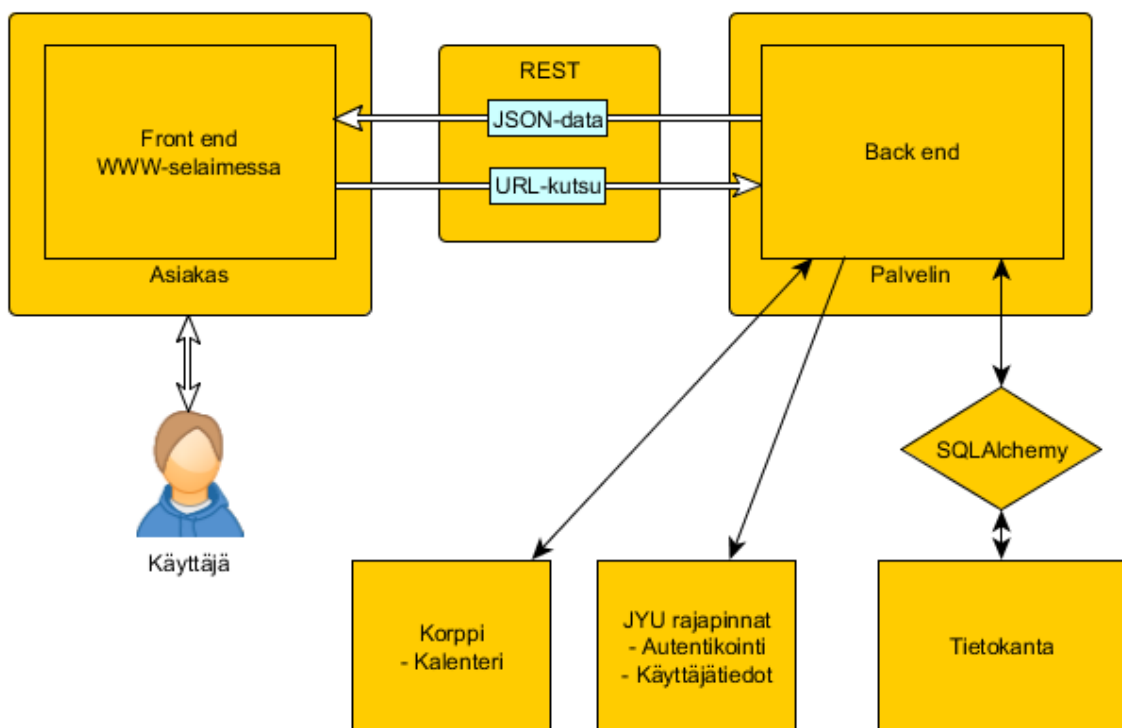
Kuva 3.26: Jyväskylän yliopiston OAuth-palvelu pyytää lupaa jakaa Kepler-järjestelmälle tietoja.

4 Sovelluksen kokonaisrakenne ja rajapinnat

Luvussa kuvataan tietojärjestelmän kokonaisrakennetta, käytettyjä ulkopuolisia komponentteja ?? sekä käytettyjä ulkopuolisia rajapintoja ?. Kepler-projektissa kehitetyn käyttöliittymän käyttäjille tarjoamat tiedot ja toiminnot on kuvattu tarkemmin vaatimusmäärittelyssä [9]. Sovelluksen luokat ja niiden rajapinnat kuvataan yksityiskohtaisesti luokkadokumentaatioissa [5] ja [6].

4.1 Sovelluksen kokonaisrakenne

Sovelluksen kokonaisrakennetta ja rajapintoja on kuvattu kuvassa 4.1.



Kuva 4.1: Sovelluksen kokonaisrakenne.

Sovellus jakautuu seuraaviin kokonaisuuksiin:

Front end on WWW-selaimessa pyörivä osa sovelluksesta, eli Keplerin

	käyttöliittymä.
REST	on REST-periaatteita noudattava rajapinta, jonka välityksellä back end ja front end keskustelevat.
Back end	on palvelimella pyörivä osa sovelluksesta, joka suorittaa toiminnot ja tekee toimintoihin liittyvät tarkistukset.
SQLAlchemy	on kolmannen osapuolen komponentti, joka pitää huolen back endin ja tietokannan välisestä kommunikaatiosta.
Tietokanta	sisältää tietojärjestelmän varastoimat ja tarvitsemat tiedot.
Korppi-kalenteri	sisältää Jyväskylän yliopiston opiskelijoiden ja henkilökunnan kalenteritietoja.
JYU-rajapinnat	ovat Jyväskylän yliopiston OAuth-palvelun tarjoamia rajapintoja, jotka mahdollistavat käyttäjän autentikoinnin ja käyttäjätietojen noutamisen.

Kuvassa 4.1 esitettävistä kokonaisuuksista Korppi-kalenteri ei toteutunut Kepler-projektin puitteissa. Lisätietoa tästä löytyy luvusta ??.

Sovelluksen yleisessä käyttötapauksessa toiminta alkaa siitä, että käyttäjä navigoi selaimellaan haluamaansa näkymään. Tämän jälkeen hän suorittaa selain-taan käyttäen jonkin (Keplerin) toiminnon, jolloin järjestelmän palvelimelle lähtee HTTP POST-pyyntö, jonka body osassa on mahdollisesti joitain parametreja JSON-muodossa. Palvelin suorittaa käyttäjän pyytämän toiminnon, mikäli pyyntö läpäisee erinäiset tarkistukset. Palvelin joutuu todennäköisesti käsittelemään tietokannassa olevaa tietoa suorittaaksen toiminnon. Tämän jälkeen palvelin lähettää käyttäjän selaimelle vastauksen JSON-muodossa. Käyttäjän selaimessa pyörivä Kepler-järjestelmän JavaScript käsittelee saamansa JSON:in ja esittää sen sisällön käyttäjän selaimessa.

4.2 Ulkopuoliset komponentit

Tietojärjestelmä hyödyntää ulkopuolisia komponentteja seuraavista kirjastoista ja ohjelmista:

Beaker	on Python WWW-sovelluksia tukeva kirjasto sessionhallintaan [12].
Bootstrap	on HTML-, CSS- ja JavaScript-kieliä tukeva kirjasto skaalautuvien WWW-sivujen luontiin [13].
dateutil	on Pythonin omaa <code>DateTime</code> -tyyppiä laajentava kirjasto [14].
docutils	on Python-kirjasto, joka muuttaa dokumentaatiota suosittuihin formaatteihin (esimerkiksi HTML, XML ja \LaTeX) [15].
FullCalendar	on jQuery:n laajennus, jolla voi helposti luoda ja muokata WWW-sivulla esitettävää kalenteria [16].
jQuery	on JavaScript-kirjasto, joka laajentaa JavaScriptin perustoinnallisuuksia ja helpottaa niiden käyttöä [17].
icalendar	on Python-kirjasto iCalendar-formaatissa olevan (kalenteri)datan lukemiseen, käsittelyyn ja tuottamiseen [18].
Pyramid	on Python WWW-sovelluskehys, jonka päälle Kepler-järjestelmä on rakennettu [19].
pyramid_beaker	on Python-kirjasto, jolla Beakerin saa helposti integroitua Pyramidiin. Kirjasto ei ole enää aktiivisessa kehityksessä [20].
pyramid_mailer	on Python-kirjasto, jonka avulla Pyramidiin saa helposti lisättyä mahdollisuuden lähettää sähköpostiviestejä [21].
Requests	on Python kirjasto HTTP- ja HTTPS-pyyntöjen tekemiseen. Se on tehty Pythonin omaa kirjastoa (<code>urllib2</code>) helpommin käytettäväksi [22].
SQLAlchemy	on Python-kirjasto, joka helpottaa relaatiotietokantojen käyttöä [23].

Järjestelmän kehityksen aikana siinä käytettiin myös kolmea muuta ulkopuolista kirjastoa, joista sitten luovuttiin.

`jQuery cookie` on jQuery:n laajennus selaimen evästeiden (cookies) hallintaa varten. Tätä käytettiin kehityksen aikana käyttäjän kielenvalinnassa, jotta pääsimme lisäämään ja testaamaan käännöksiä, ennen kuin ehdimme toteuttaa oman käännösratkaisumme.

`Tablesorter` on JavaScript-kirjasto taulukkomuotoisen datan lajitteluun HTML-taulukossa. Siitä luovuttiin, koska se ei toiminut haluamallamme tavalla (lajittelu-perusteen valinta ei ollut tarpeeksi kattava).

`requests_oauthlib` on `Requests`-kirjastoon pohjautuva OAuth-toteutus. Sitä käytettiin, kun Jyväskylän yliopiston tarjoama OAuth-rajapinta ei meinannut alkaa toimimaan (myöhemmin paljastui, että kyseessä oli puuttuvat käyttöoikeudet). Jos virhe olisi ollut omassa OAuth-totetuksessamme, olisi sen pitänyt toimia `requests_oauthlib`:llä. `requests_oauthlib`:iä ei kuitenkaan saanut helposti tuottamaan Jyväskylän yliopiston OAuth-rajapinnan vaatiman muotoista dataa, joten siitä luovuttiin.

Järjestelmän kehityksen aikana mietittiin `pyramid_beaker`:stä luopumista. Tämä johtui siitä, että se ei ole enää aktiivisessa kehityksessä. Myös `Requests`-kirjastosta luopumista harkittiin. Tähän ei varsinaista syytä ollut muuta kuin se, että Kepler-järjestelmän riippuvuutta ulkopuolisista kirjastoista saataisiin vähennettyä. `Requests` on kuitenkin ollut jo pidemmän aikaa vakaassa kehityksessä ja sen käyttöön löytyy helposti apua, joten siitä ei tarvitse luopua.

4.3 Olennaisimmat muutokset toteutusratkaisuissa

Sovelluksen kokonaisrakenteeseen ei tehty muutoksia projektin aikana. Suurimmat muutokset olivat sovelluksen sisäiseen toimintalogiikkaan liittyvät refaktoroinnit. Nämä refaktoroinnit koostuivat usein käytettyjen toimintojen muuttamisesta käyttämään erinäisiä apuluokkia.

Refaktorointi tapahtui esimerkiksi virheviestien esittämisen yhteydessä. Ne muutettiin pelkkien merkkijonojen palauttamisen sijasta käyttämään erilaisia apuluokkia, jolloin viesteistä saadaan yhdenmukaisia. Virheviestien esittäminen refaktorointiin Kepler-projektin aikana pariinkin kertaan ja nyt virheviestit voi (pääasiassa) esittää helposti ja yhtenevästi käyttöliittymässä.

4.4 Muutettavat asetukset

Käyttäjä voi muuttaa kielivalintaansa pysyvästi omista tiedoistaan, kuten luvussa ?? kuvattiin.

Keplerin ylläpitäjä voi muuttaa joitakin asetuksia, muuttamalla tiedostoa `config.py` (projektin juuresta katsoen sen tiedostopolku on `/kepler/config.py`). Alla on kuvattuna esimerkki sen sisällöstä.

```
class Config(object):
    """Config class defines the configuration values of the
    Kepler system."""

    time_slot_max_reservations_default = 6
    """The default value for the limit of reservations that
    can be made on a time slot."""

    reservation_cancel_workdays_limit = 2
    """Reservations can be canceled only if there is more than
    the specified number of workdays before the start of the
    time slot."""

    reservation_make_workdays_limit = 2
    """Reservations can be made only on time slots that are
    more than the specified workdays in the future."""

    user_group_max_reservations = 4
    """One user group can have at most the specified number of
    active reservations."""

    user_group_min_size = 2
    """Defines the minimum number of members a user group can
    have."""

    user_group_max_size = 2
    """Defines the maximum number of members a user group can
    have."""

    default_language_id = constants.Language.finnish_id.value
    """The ID of the default language of the system."""
```

Lisäksi on olemassa asetustiedostot `development.ini` ja `production.ini` (jotka löytyvät järjestelmän kansiorakenteen juuresta). Näistä löytyy esimerkiksi säh-

köpostiin ja käytettävään palvelin ohjelmistoon littyviä asetukset. Nämä asetukset astuvat voimaan, kun palvelu käynnistyy. `development.ini` on tarkoitettu kehittämisskäyttöön eikä sitä saa missään nimessä käyttää tuotantoympäristössä! Alla on esitetty `production.ini`:n nykyinen sisältö.

```
###
# app configuration
# http://docs.pylonsproject.org/projects/pyramid/en/1.5-branch/narr/
# environment.html
###

[app:main]
use = egg:Kepler

pyramid.reload_templates = false
pyramid.debug_authorization = false
pyramid.debug_notfound = false
pyramid.debug_routematch = false
pyramid.default_locale_name = en
pyramid.includes =
    pyramid_tm
    pyramid_chameleon
    pyramid_beaker
    pyramid_mailer

sqlalchemy.url = sqlite:///%(here)s/Kepler.sqlite

###
# Beaker session config
###

# The session invalidation timeout in seconds. (900s = 15min)
session.timeout = 900
# Tells web browser that the session ID cookie is to be sent only over
# secure connection.
session.secure = true
# Used with the HMAC to ensure session integrity.
```

```
session.secret = 'b1203525208e0b0934b43aeb0c5ccf00ee832071'
```

```
[server:main]
use = egg:waitress#main
host = 127.0.0.1
port = 6543
# Add trusted_proxy IP to allow X_FORWARDED_PROTO header to
# override default URL scheme
# 127.0.0.1 = IP of localhost
trusted_proxy = 127.0.0.1
# url scheme when using secure HTTPS connection :
url_scheme = 'https'
```

```
###
# logging configuration
# http://docs.pylonsproject.org/projects/pyramid/en/1.5-branch/narr/
###
```

```
[loggers]
keys = root, kepler, sqlalchemy
```

```
[handlers]
keys = console
```

```
[formatters]
keys = generic
```

```
[logger_root]
level = WARN
handlers = console
```

```
[logger_kepler]
level = WARN
handlers =
qualname = kepler
```

```
[logger_sqlalchemy]
level = WARN
handlers =
qualname = sqlalchemy.engine
# "level = INFO" logs SQL queries.
# "level = DEBUG" logs SQL queries and results.
# "level = WARN" logs neither.  (Recommended for production systems.)

[handler_console]
class = StreamHandler
args = (sys.stderr,)
level = NOTSET
formatter = generic

[formatter_generic]
format = %(asctime)s %(levelname)-5.5s [%(name)s][%(threadName)s] %(message)s
```

4.5 Ulkoiset rajapinnat

Kepler-projektissa käytetään Jyväskylän yliopiston tietojärjestelmien rajapintoja.

4.5.1 Korppi-kalenteri

Alun perin suunnitelmissa oli tuoda Kepler-järjestelmän kalentereihin Korppi-kalenterin tapahtumat ja viedä Kepler-järjestelmän tapahtumat Korppi-kalenteriin. Tämä jäi kuitenkin toteuttamatta yhteistyön Korpin kanssa viivästyessä. Korppi-kehittäjät tekivät OAuthin kautta toimivan esimerkkitoteutuksen, mutta varsinainen toteutus jäi vielä tulevaisuuteen.

Keplerissä on lähdetty työstämään esimerkkitoteutuksen mallidatan (iCalendar muotoista) parsimista ja käyttöä. HTTP-kutsuihin käytetty `Requests`-kirjasto ei kuitenkaan toimi kyseisessä tapauksessa odotetulla tavalla ja kyseinen ominaisuus pitäisi toteuttaa Pythonin valmiilla `urllib2`-kirjastolla. Koska ongelma ilmeni myöhäisessä vaiheessa projektia, se sovittiin tilaajan kanssa jätettävän jatkokehitykseen. iCalendar muotoisen datan parsiminen ja tuottaminen kyllä toimivat.

4.5.2 OAuth

Kepler-järjestelmään kirjaututaan Jyväskylän yliopiston tarjoaman OAuth2-rajapinnan kautta. Näin Kepler-järjestelmän ei itse tarvita huolehtia autentikoinnista.

OAuth-rajapinnan kautta Kepler myös saa käyttäjistä tietoa (esimerkiksi nimet ja sähköpostiosoitteen), joten käyttäjän ei tarvitse erikseen täyttää ja päivittää tietojaan Kepler-järjestelmään.

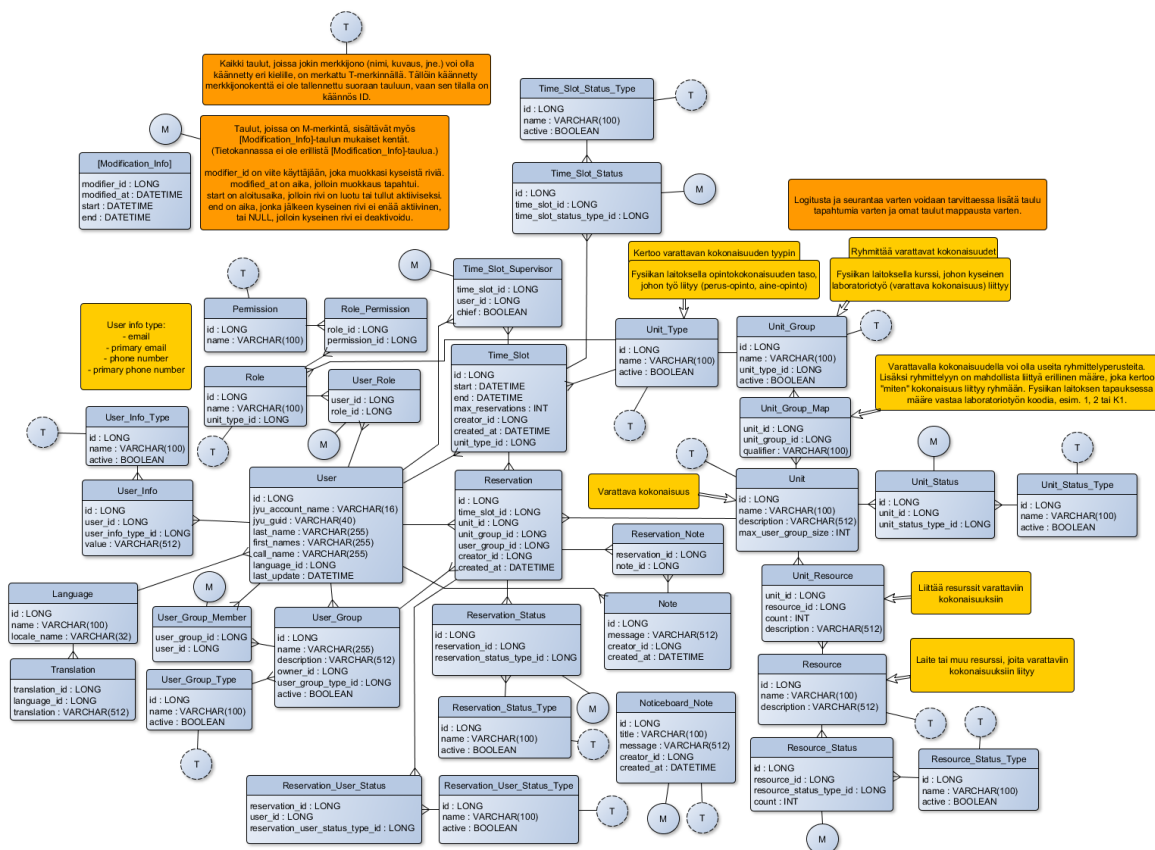
Jyväskylän yliopiston OAuth-rajapinnasta löytyy tarkempaa tietoa sen ohjeista ??. Tämän lisäksi kannattaa tutustua OAuthin viralliseen RFC:hen ??.

Kepler-järjestelmään OAuth-autentikoinnin ja tietojen haun toteutus on tehty itse, sillä mikään löydetyistä valmiista (ja yhä aktiivisesti päivitettyistä) kirjastoista ei olisi toiminut ilman muutoksia. Jyväskylän yliopiston OAuth-rajapinnalla on erikoisvaatimus, jota RFC:ssä ei ole. Se vaatii, että `authorization code grant flow:ta` käytettäessä `client_id` sisällytetään POST-pyyynnön `body`-osaan, kun rajapinnalta pyydetään `tokenia`. Kepler-järjestelmä käyttää OAuth-autentikoitiin `authorization code grant flow:ta`.

5 Tietokannan rakenne

5.1 Kepler-järjestelmän tietokannasta

Koska tietojärjestelmästä oli tavoitteena tehdä yleiskäyttöinen, piti tietokannastakin tehdä sellainen. Oheessa on esitetty kuva 5.1, joka kuvaa tietokannan rakennetta.



Kuva 5.1: Tietokannan rakenne.

Tietokannassa toistuu rakenne, jossa "perustaululle" on olemassa myös Status- ja Type-taulut. Status-taulu kuvaa aina perustaulun elementin jonkinlaista tilaa. Type-taulu taas antaa Status-taulun kuvaamalle tilalle nimen. Status- ja Type-taulut on eritelty erillisiksi rakenteiksi sen takia, että eri tilojen nimiä (ja mahdollisesti muita ominaisuuksia, mikäli sellaisia tulevaisuudessa tulee) olisi helppo muuttaa hajottamatta mitään olemassa olevaa toimintalogiikkaa.

Tietokannan kuvassa 5.1 on käytetty merkintöjä T ja M kuvan selkeyttämiseksi (samasta syystä myös monia taulujen välisiä suhteita on jätetty merkitsemättä). Merkintä M tarkoittaa, että tauluun sisältyy kentät `modifier_id`, `modified_at`, `start` ja `end`, kuten kuvassa 5.1 näkyvässä [ModificationInfo] taulussa näkyy. Merkintä T taas puolestaan tarkoittaa, että sen omaavassa taulussa ei oikeasti säilötäkään merkkijonoja, vaan niiden sijaan säilötään `translation_id`, joka viittaa haluttuun merkkijonoon Translation-taulussa.

Koska tietokantaa voi olla vaikea ymmärtää, käydään jokaisen taulun tarkoitus läpi alla. Taulut käydään läpi kokonaisuuksittain. Kuvassa 5.1 olevat alaviivat jätetään toistamatta, koska ne eivät ole käytössä tietojärjestelmän toteutuksessakaan.

5.2 Permission

Taulu sisältää eri toimintojen vaatimat valtuudet.

id Valtuuden yksilöivä tunniste.

name Valtuuden nimi.

5.3 RolePermission

Taulu liittää valtuudet rooleihin.

role_id Roolin yksilöivä tunniste.

permission_id Valtuuden yksilöivä tunniste.

5.4 Role

Taulu sisältää tietojärjestelmän eri käyttäjäroolit.

id Roolin yksilöivä tunniste.

name Roolin nimi.

unit_type_id Kertoo, minkä tyyppiseen varattavaan yksikköön (esimerkiksi perus- vai aineopinnot) rooli kuuluu.

5.5 UserRole

Taulu sisältää tiedon siitä, mitä rooleja kullakin käyttäjällä on.

user_id Käyttäjän yksilöivä tunniste.

role_id Roolin yksilöivä tunniste.

5.6 User

Taulu sisältää tietoja tietojärjestelmän käyttäjistä. `last_update`-kenttää ei nykyisellään käytetä.

id Käyttäjän yksilöivä tunniste.

jyu_account_name Käyttäjän Jyväskylän yliopiston tietojärjestelmien käyttäjätunnus.

jyu_guid Käyttäjän Jyväskylän yliopiston tietojärjestelmissä yksilöivä tunniste.

last_name Käyttäjän sukunimi.

first_names Käyttäjän etunimet.

call_name Käyttäjän kutsumanimi.

language_id Käyttäjän oletuskielivalinta (Kepler-järjestelmässä).

last_update Tieto siitä, milloin käyttäjän tietoja on viimeksi päivitetty (ei käytetä nykyisessä toteutuksessa).

5.7 UserInfo

Taulu sisältää käyttäjään liitettävää tietoa. Sen merkittävin ero `User`-tauluun on se, että sen sisältämät tiedot ovat luonteeltaan sellaisia, jotka muuttuvat useammin. Nykyisellään `UserInfo`-taulu sisältää vain käyttäjän yhteystietoja.

id	Käyttäjän tiedon yksilöivä tunniste.
user_id	Käyttäjän yksilöivä tunniste.
user_info_type_id	Käyttäjän tiedon tyyppin yksilöivä tunniste.
value	Käyttäjistä säilötty tieto.

5.8 UserInfoType

Taulu sisältää tiedon `UserInfo`-taulun rivin sisältämän tiedon tyyplistä.

id	Käyttäjän tiedon tyyppin yksilöivä tunniste.
name	Käyttäjän tiedon tyyppin nimi.
active	Kertoo, onko käyttäjän tiedon tyyppi aktiivisena järjestelmässä.

5.9 Language

Taulu sisältää tietoa järjestelmän tukemista kielistä.

id	Kielen yksilöivä tunniste.
name	Kielen nimi.
locale_name	Kielen tunnistava <code>locale</code> (esimerkiksi <code>fi-FI</code>).

5.10 Translation

Taulu sisältää kaikki tietojärjestelmän sisältämän tietojen ja virheviestien käännökset.

translation_id Merkkijonon käännöksen yksilöivä tunniste.

language_id Kielen yksilöivä tunniste.

translation Merkkijonon käännös.

5.11 UserGroup

Taulu sisältää tietoa järjestelmässä olevista käyttäjien muodostamista ryhmistä.

id Käyttäjärühmän yksilöivä tunniste.

name Käyttäjärühmän nimi.

description Käyttäjärühmän kuvaus (ei käytössä nykyisessä toteutuksessa).

owner_id Käyttäjärühmän omistajan (nykyisellään ryhmän luonut käyttäjä) yksilöivä tunniste.

user_group_type_id Käyttäjärühmän tyyppin yksilöivä tunniste.

active Kertoo, onko käyttäjärühmä aktiivinen järjestelmässä.

5.12 UserGroupMember

Taulu sisältää kuhunkin ryhmään kuuluvat käyttäjät.

user_group_id Käyttäjärühmän yksilöivä tunniste.

user_id Käyttäjän yksilöivä tunniste.

5.13 UserGroupType

Taulu sisältää tiedon ryhmän tyypistä (esimerkiksi, henkilökohtainen ryhmä tai opiskelijoiden ryhmä).

id	Käyttäjärühmän tyypin yksilöivä tunniste.
name	Käyttäjärühmän tyypin nimi.
active	Kertoo, onko käyttäjärühmin tyyppi aktiivisena järjestelmässä.

5.14 TimeSlot

Taulu sisältää tietoa järjestelmässä olevista ajanjaksoista (esimerkiksi fysiikan laitoksella laboratoriovuoro on ajanjakso), joiden ajalle voi tehdä varauksia.

id	Ajanjakson yksilöivä tunniste.
start	Ajanjakson aloitus aika.
end	Ajanjakson lopetus aika.
max_reservations	Maksimi määrä varauksia, mitä ajanjaksolle sallitaan.
creator_id	Ajanjakson luoneen käyttäjän yksilöivä tunniste.
created_at	Ajanhetki, jolloin ajanjakso luotiin.
unit_type_id	Niiden yksiköiden tyyppi, joita ajanjaksolla oleviin varauksiin voi liittää.

5.15 TimeSlotSupervisor

Taulu sisältää tiedon ajanjaksojen valvojana tai ohjaajana toimivista henkilöistä. Näistä henkilöistä tiedetään myös, kuka tai ketkä heistä ovat vastuussa.

time_slot_id	Ajanjakson yksilöivä tunniste.
user_id	Käyttäjän, eli valvojan, yksilöivä tunniste.
chief	Kertoo, onko kyseinen valvoja vastuussa ajanjaksosta.

5.16 TimeSlotStatus

Taulu liittää yhteen ajanjakson ja siihen liittyvän tilan.

id	Ajanjakson tilan yksilöivä tunniste.
time_slot_id	Ajanjakson yksilöivä tunniste.
time_slot_status_type_id	Ajanjakson tilan tyyppin yksilöivä tunniste.

5.17 TimeSlotStatusType

Taulu sisältää tilat, joissa ajanjakso voi olla (esimerkiksi suunnitteilla oleva tai varmistettu). Taulun sisältöä ei oteta nykyisen toteutuksen toimintalogiikassa huomioon.

id	Ajanjakson tilan tyyppin yksilöivä tunniste.
name	Ajanjakson tilan tyyppin nimi.
active	Kertoo, onko kyseinen ajanjakson tilan tyyppi yhä käytössä järjestelmässä.

5.18 Reservation

Taulu sisältää tiedon järjestelmän ajanjaksoille tehdyistä varauksista ja niihin liittyvistä tiedoista.

id	Varauksen yksilöivä tunniste.
name	Ajanjakson yksilöivä tunniste.
unit_id	Varattavan yksikön yksilöivä tunniste.
unit_group_id	Varattavan yksikön kokonaisuuden (esimerkiksi fysiikan kurssi) yksilöivä tunniste.
user_group_id	Käyttäjärühmän, joka varauksen tekee, yksilöivä tunniste.

creator_id	Varauksen tehneen käyttäjän yksilöivä tunniste.
created_at	Varauksen tekemisen ajankohta.

5.19 ReservationStatus

Taulu liittää yhteen varauksen ja siihen liittyvän tilan.

id	Varauksen tilan yksilöivä tunniste.
reservation_id	Varauksen yksilöivä tunniste.
reservation_status_type_id	Varauksen tilan tyyppin yksilöivä tunniste.

5.20 ReservationStatusType

Taulu sisältää tilat, joissa varaus voi olla (esimerkiksi pääkäyttäjän peruma tai aktiivinen).

id	Resurssin tilan tyyppin yksilöivä tunniste.
name	Resurssin tilan tyyppin nimi.
active	Kertoo, onko kyseinen varauksen tilan tyyppi yhä käytössä järjestelmässä.

5.21 ReservationUserStatus

Taulu liittää yhteen varauksen ja siihen liittyvän käyttäjän tilan.

reservation_id	Varauksen yksilöivä tunniste.
user_id	Käyttäjän yksilöivä tunniste.
reservation_user_status_type_id	Varauksen käyttäjän tilan tyyppin yksilöivä tunniste.

5.22 ReservationUserStatusType

Taulu sisältää varaukseen osallistuneen henkilön mahdolliset tilat (esimerkiksi paikalla tai poissa).

- id** Varauksen käyttäjän tilan tyyppin yksilöivä tunniste.
- name** Varauksen käyttäjän tilan tyyppin nimi (esimerkiksi oli paikalla tai ei ollut paikalla).
- active** Kertoo, onko kyseinen varauksen käyttäjän tilan tyyppi yhä käytössä järjestelmässä.

5.23 ReservationNote

Taulu liittyy yhteen varauksen ja siihen liittyvän viestin.

- reservation_id** Varauksen yksilöivä tunniste.
- note_id** Viestin yksilöivä tunniste.

5.24 Note

Taulu sisältää kaikki järjestelmään kirjatut viestit.

- id** Viestin yksilöivä tunniste.
- message** Viestin sisältö.
- creator_id** Viestin luoneen käyttäjän yksilöivä tunniste.
- created_at** Viestin luonnin ajankohta.

5.25 NoticeboardNote

Taulu sisältää tiedon ilmoitustaulun ilmoituksista ja niihin liittyvistä tiedoista.

id	Ilmoituksen yksilöivä tunniste.
title	Ilmoituksen otsikko.
message	Ilmoituksen sisältö.
creator_id	Ilmoituksen luoneen käyttäjän yksilöivä tunniste.
created_at	Ilmoituksen luonnin ajankohta.

5.26 Unit

Taulu sisältää varattavat yksiköt ja niihin liittyvät tiedot (esimerkki tällaisesta yksiköstä on fysiikan laboratoriotyö). Varattavat yksiköt muodostuvat resursseista.

id	Varattavan yksikön yksilöivä tunniste.
name	Varattavan yksikön nimi.
description	Varattavan yksikön kuvaus.
max_user_group_size	Yksikön varaavan käyttäjäryhmän maksimi koko.

5.27 UnitGroupMap

Taulu liittää yhteen varattavat yksiköt ja niiden muodostamat yksiköiden ryhmät. Tämän lisäksi taulu sisältää varattavan yksikön ja siihen liittyvän kokonaisuuden yksilöivän tunnisteiden.

unit_id	Varattavan yksikön yksilöivä tunniste.
unit_group_id	Varattavien yksiköiden kokonaisuuden yksilöivä tunniste.
qualifier	Varattavan yksikön ja siihen liittyvän kokonaisuuden yksilöivä tunniste.

5.28 UnitGroup

Taulu sisältää tiedon siitä, mihin varattavien yksiköiden kokonaisuuteen (esimerkiksi kurssin vaatimat laboratoriotyöt) mikäkin yksikkö kuuluu.

id	Varattavien yksiköiden kokonaisuuden yksilöivä tunniste.
name	Varattavien yksiköiden kokonaisuuden nimi.
unit_type_id	Varattavien yksiköiden tyyppi (esimerkiksi perus- tai aineopinnot).
active	Kertoo, onko kyseinen varattavien yksiköiden kokonaisuus yhä käytössä järjestelmässä.

5.29 UnitType

Taulu sisältää tiedon varattavan yksikön tyypistä (esimerkiksi perus- tai aineopinnot).

id	Varattavien yksiköiden tyypin yksilöivä tunniste.
name	Varattavien yksiköiden tyypin nimi.
active	Kertoo, onko kyseinen varattavien yksiköiden tyyppi yhä käytössä järjestelmässä.

5.30 UnitStatus

Taulu liittää yhteen varattavan yksikön ja siihen liittyvän tilan.

id	Varattavan yksikön tilan yksilöivä tunniste.
unit_id	Varattavan yksikön yksilöivä tunniste.
unit_status_type_id	Varattavan yksikön tilan tyypin yksilöivä tunniste.

5.31 UnitStatusType

Taulu sisältää tilat, joissa varattava yksiköt voivat olla. Taulun sisältöä ei oteta nykyisellään huomioon järjestelmän toimintalogiikassa.

id	Varattavan yksikön tilan tyyppin yksilöivä tunniste.
name	Varattavan yksikön tilan tyyppin nimi.
active	Kertoo, onko kyseinen varattavan yksikön tilan tyyppi yhä käytössä järjestelmässä.

5.32 UnitResource

Taulu liittää resurssit varattaviin yksiköihin (esimerkiksi mittanauhat laboratorio-työhön).

unit_id	Varattavan yksikön yksilöivä tunniste.
resource_id	Resurssin yksilöivä tunniste.
count	Varattavaan yksikköön vaadittujen resurssien määrä.
description	Kuvaus siitä, miksi resurssit liittyvät varattavaan yksikköön.

5.33 Resource

Taulu sisältää tietoa järjestelmässä olevista resursseista (esimerkiksi mittanauhat), joista muodostetaan varattavia yksiköitä.

id	Resurssin yksilöivä tunniste.
name	Resurssin nimi.
description	Resurssin kuvaus.

5.34 ResourceStatus

Taulu liittää resurssit ja niihin liittyvän tilan. Taulu myös sisältää tiedon siitä, kuinka monta resurssia kussakin tilassa on.

id	Resurssin tilan yksilöivä tunniste.
resource_id	Resurssin yksilöivä tunniste.
resource_status_type_id	Resurssin tilan tyyppin yksilöivä tunniste.
count	Kertoo, kuinka monta kyseistä resurssia on kyseisessä tilassa.

5.35 ResourceStatusType

Taulu sisältää tilat, joissa resurssi voi olla.

id	Resurssin tilan tyyppin yksilöivä tunniste.
name	Resurssin tilan tyyppin nimi.
active	Kertoo, onko kyseinen resurssin tilan tyyppi yhä käytössä järjestelmässä.

6 Tavoitteiden toteutuminen

Luvussa kuvataan vaatimusten toteutumista sekä heikkoja ja puutteellisia toteutusratkaisuja. Sovelluksen vaatimukset ja niiden toteutuminen on kuvattu vaatimusmäärittelyssä [9].

Vaatimusmäärittelyn ensimmäisen prioriteetin 38:sta vaatimuksesta toteutui kokonaan 32. Yhtä vaatimusta (8.2.4 50 yhtäaikaisen käyttäjän palveleminen) ei testattu, joten se lasketaan toteutumattomaksi.

Rajallisen aikataulun vuoksi sovellukseen jäi myös heikkoja ja puutteellisia toteutusratkaisuja.

Ohjelmointikäytännöt ja niiden toteutuminen on kuvattu projektiraportin [7] luvussa 5.5.

6.1 Vaatimusten toteutuminen

Vaatimusten toteutuminen vaatimuskohtaisesti on kuvattu vaatimusmäärittelyssä [9]. Pakollisista 38 vaatimuksesta toteutui kokonaan 32 vaatimusta ja kaksi toteutettiin osittain. Osittain toteutettu pakollinen vaatimus 7.7.9 liittyy tarkastukseen laboratoriotyötä (tietokannassa `Unit`-taulussa) poistettaessa, silloin kun se kuuluu kurssiin (tietokannassa `UnitGroup`-taulussa). Osittain toteutettu pakollinen vaatimus 8.2.3 liittyy järjestelmään vastustuskykyyn CSRF-hyökkäystä (cross site request forgery) vastaan, sillä tätä vastaan on varauduttu ainakin paikoittain, mutta järjestelmällisesti sitä ei ole testattu.

Tärkeistä 27 vaatimuksesta toteutettiin 15. Osittain toteutettiin seuraavat vaatimukset:

7.5.4

Käyttäjä pystyy lisäämään käännöksen ilmoituksen otsikolle ja sisällölle. Tämän pystyy tekemään, muttei luontevalla tavalla. Tämän takia sitä pidetään osittain toteutettuna.

7.9.1

(Pää)käyttäjä pystyy lisäämään käyttäjille uusia rooleja. Toteuttuna testattomana palvelinkoodissa.

- 7.9.2** Käyttäjä pystyy merkitsemään käyttäjän roolin poistetuksi. Toteuttuna testaamattomana palvelinkoodissa.
- 8.1.4** Kalenterinäköymän tuontiin Korpista käytetään iCalendar-muotoa. Valittu toteutusratkaisu ei toimi. Sen saisi toimimaan, kun kirjoittaisi sen uudelleen käyttäen `urllib2:ta Requestsin` sijaan.

Mahdollisista 40 vaatimuksesta toteutettiin 19 ja osittain toteutettiin 5 vaatimusta. Toteuttamatta jäi 16 mahdollista vaatimusta. 20 ideatason vaatimusta sovittiin jo alussa projektin ulkopuolelle. Vaatimusmäärittelyyn ei kirjattu prioriteetilla *Ei toteuteta* yhtään vaatimusta.

Vaatimusten toteutumisesta kerrotaan myös vaatimusmäärittelyn [9] luvussa 9.

6.2 Suoritetut testaukset ja niiden tulokset

Sovelluksen toteutusvaiheessa jokainen ryhmän jäsen suoritti manuaalista testausta, testaten tekemiään muutoksia. Testausta vaikeutti se, että käyttöliittymän ja palvelimen toteutus riippuivat olennaisesti toisistaan, jolloin molempien osien piti olla valmiina testausta varten. Näin ollen esimerkiksi palvelinkoodin virheitä ei yleensä huomattu ennen kuin sitä käyttävä käyttöliittymän osa valmistui.

Projektin lopussa Mikko Kuhno laati testausraportin [4], joka kattoi suoritetut testitapaukset. Järjestelmätestaus suoritettiin Windows-ympäristössä Internet Explorer (versio 11.0.9600.17801) selaimella ja Linux-ympäristössä ja Mozilla Firefox (versio 35.0.1) selaimella. Testauskertoja ei suoritettu todellisessa käyttötilanteessa.

Projektin aikana järjestettiin käytettävyytestauspäivä, jonka aikana käytettävyyssiantuntija Johanna Silvennoinen antoi palautetta sovelluksen käytettävyydestä.

Projektin loppupuolella Anu Koskela suunnitteli ja suoritti järjestelmän käytettävyytestauksen. Käytettävyytestauksessa käyttäjinä oli Jyväskylän fysiikan laitoksen opiskelijoita, joista osa toimi laboratoriovuorojen ohjaajina ja osa oli tavallisia opiskelijoita. Käytettävyytestauksessa havaitut ongelmat huomioitiin sovelluksen toteutuksessa. Käytettävyytestauksesta tehtiin suunnitelma [2] ja muistio [3].

Projektin aikana sovelluksen eri versioita koekäyttivät vastaava ohjaaja Jukka-Pekka Santanen, tekninen ohjaaja Petri Partanen, sekä tilaajan edustajat Sakari Juutinen ja Panu Rahkila. Erityisesti Jukka-Pekka Santanen ja tilaajan edustajat antoivat sovelluksesta palautetta.

6.3 Heikot ja puutteelliset toteutusratkaisut

Käyttöliittymän JavaScriptissä syötetään paikoittain sivulle suoraan HTML-koodia. Tämä pitäisi korvata oikeaoppisesti JavaScriptillä.

Käyttöliittymässä käytetään vielä paikoin JavaScriptin alertteja ilmoitusten yhteydessä. Nämä olisi syytä korvata viher- ja punataustaisilla teksteillä (niin kuin monessa kohtaa on jo tehtykin).

Kun selainikkunaa pienennetään (tai Kepleriä käytetään laitteella, jonka näytön resoluutio on pieni), järjestelmän nimen yhteydessä oleva järjestelmän kuvaus ei rivity oikein ja jää osittain navigointipalkin alle piiloon.

Järjestelmässä pitäisi käyttää kaikkialla yhtenevää väriteemaa. Nykyisellään asiaa ei ole perinpohjaisesti selvitetty, joten värit voivat oli eri näkymissä eriäviä.

Kun käyttäjän sessio vanhenee (ja hänet kirjataan automaattisesti ulos), pitäisi tilanteesta antaa käyttäjälle selkeä ilmoitus hänen yrittäessään tehdä jotain (pelkän kirjautumisnäkymän uudelleenohjaamisen sijasta).

HTML:ään tulisi lisätä selitteet (eli lähinnä `title-tagit`) erilaisia lukijalaitteita varten, jotta Kepler olisi käytettävämpi erikoisissa käyttötapauksissa (esimerkiksi sokealle ihmiselle).

Uutta vuoroa luotaessa pitäisi pystyä määrittelemään, onko vuoro vasta suunnitella vai tuleeko se toteutumaan (nyt käyttöliittymässä on uutta vuoroa luotaessa *Vuoron status*-valinta, jonka valinnalla ei ole vaikutusta järjestelmän toimintalogiikkaan).

Käyttäjäroolienhallinta on toteutettuna palvelimen päässä (vaikkakin testaamattomana), mutta sille pitäisi vielä tehdä käyttöliittymä. Sitä varten oleva käyttöliittymä ei nykyisellään sisällä mitään toiminnallisuutta eikä se ole käytettävyydeltään hyvä (joten se pitäisi tehdä kokonaan uudestaan).

Palvelimen koodissa luotetaan siihen, että tietokannasta tuleva data ei aiheuta tietoturvaongelmia. Käyttäjän syötettä ei kyllä pitäisi päästä (XSS:ää varten) käsittelemättömänä tietokantaan, mutta tietokannasta tulevaan dataan ei silti ole viisasta sokeasti luottaa ja sillekin olisi syytä suorittaa jonkinlaisia operaatiota ja tarkistuksia.

Järjestelmä ei tee minkäänlaisia lokeja WWW-palvelimen omien lokien lisäksi. Tämä olisi syytä korjata. Jos lokit haluaa liittää tietokantaan, voisi ne toteuttaa esimerkiksi samantyyppisellä ratkaisulla kuin millä merkkijonojen käännökset on nykyisessä järjestelmässä toteutettu (`Translation`-taulu).

Järjestelmä ei pidä yllä kovin kattavia historiatietoja. Joistain tauluista ei löydy minkäänlaisia historiatietoja. Niistä tauluista, joista historiatietoja löytyy, koskettavat ne vain viimeisintä muutosta. Kattavammista historiatiedoista olisi hyötyä ylläpidossa ja niistä voisi tehdä tilastoja (esimerkiksi siitä, kuinka moni vuorolle ilmoittanut sinne keskimäärin saapuu).

OAuth-toteutuksen virheenkäsittely ei ole kovin sulavaa tai kattavaa. Virheviestejä ei myöskään lähetetä muulle järjestelmälle yhtenevällä tavalla.

Sähköpostiviestit tulisi testata sähköpostipalvelimen kanssa (järjestelmää kehittäessä niitä on testattu vain paikallisesti).

Varattavien töiden (varattavien yksiköiden, tietokannassa `Unit`-taulu), resurssien (tietokannassa `Resource`-taulu) ja kurssien (tietokannassa `UnitGroup`-taulu) poiston yhteydestä puuttuvat kaikki mielekkäät tarkistukset.

Nykyinen ulkopuolisen iCalendar-syötteen lukemisen toteutus ei toimi. Se pitäisi kirjoittaa uusiksi käyttäen `urllib2`:ta `Requests`in sijaan (muilta osin se toimii).

Yksi suurimmista Kepler-järjestelmän koodin ongelmista on se, että etenkin palvelinkoodissa toistetaan samoja kaavoja hyvinkin paljon. Eri toteutusten tarpeissa on kuitenkin tarpeeksi eroja, että emme tehneet Kepler-projektin puitteissa riittävän yleisen tason apuluokkia koodin modularisuuden parantamiseksi.

Etenkin erilaisten palvelinkoodin toimintalogiikkaan liittyvien tarkastusten yhteydessä olisi syytä käyttää luettavuuden nimissä aliohjelmia. Nykyisellään koodi voi olla raskasta luettavaa, ennen kuin sitä oppii lukemaan (sillä samantyyppinen koodi toistuu toiminnosta toiseen).

Pääkäyttäjän näkymissä ei ole käytettävyyteen panostettu yhtä paljon kuin muissa näkymissä (ajan puutteen ja vaatimusten priorisoinnin vuoksi).

Nykyisellään kirjautumisnäkyvä on askeettinen, joten pieni ulkonäön parantelu olisikin varmaan paikallaan ennen järjestelmän käyttöönottoa.

Tietokannan alustuskripti syöttää tietokantaan aina myös mallidataa, eli se ei sovi tuotantoympäristöön. Siitä kuitenkin saisi pienellä vaivalla poistettua tämän mallidatan syötön (mallidatan syöttävä versio kannattaa kuitenkin säilyttää kehityskäyttöä varten), jolloin sillä olisi helppo alustaa myös tuotantokäyttöön soveltuva tietokanta.

OAuth-toteutuksessa ei ole `authentication tokenin` päivittämistä (refresh) toteutettuna ollenkaan. Jyväskylän yliopiston OAuth-rajapinnan tokenit vanhenevat kuudessa tunnissa. Kepler-järjestelmä kirjaa käyttäjän automaattisesti ulos, jos hän ei ole tehnyt mitään 15 minuuttiin. Näin ollen tilannetta, jossa token pääsee vanhenemaan ei pitäisi kovin usein tapahtua. Tällaisessa tilanteessa käyttäjä saa jotain tehdessään joko virheviestin tai hänet ohjataan suoraan kirjautumissivulle. Jokainen istunto Kepler-järjestelmässä siis käyttää omaa tokeniaan.

CSS:ssä on määritelty fonttien koot pikseleissä, joten käyttöliittymän fontit eivät skaalaudu käyttäjän selaimen asetusten mukaan. Fonttien koot pitäisi muuttaa käyttämään yksiköine em:iä.

Käyttöliittymässä on paikoin, etenkin pääkäyttäjän näkymissä, ongelmia skaalautumisen kanssa, kun osa tekstistä jää muiden käyttöliittymäelementtien alle piiloon.

6.4 Sovelluksen yleiset jatkokehitysideoita

Projektin kuluessa esitettiin seuraavat kehitysideoita:

- Käyttöliittymän alapalkissa oleva `Anna palautetta`-linkki pitäisi joko poistaa tai sitä vastaava toiminnallisuus pitäisi toteuttaa.
- Käyttäjistä käytetyt nimiyhdistelmät olisi hyvä yhdenmukaistaa. Välillä käytetään etunimen ja sukunimen yhdistelmää, kun taas välillä etunimet korvataan kutsumanimellä, jota ei välttämättä ole edes olemassa.
- Vuoroja listatessa (*Omat varaukset*-, *Ohjausvuorot*- ja *Hallitse vuoroja*-näkymissä) aikavalinta *Näytetään vuorot ajalta* pitäisi joko toteuttaa tai poistaa.
- Kalenterinäkymien yhteydessä olisi hyvä esittää käytettävä kalenterin värikoodaus.

- Omat varaukset ja ohjausvuorot pitäisi saada järjestelmästä ladattua iCalendar muodossa (tämä on palvelinkoodissa toteutettuna).
- Ohjaajan ja pääkäyttäjän pitäisi kyetä lisäämään opiskelijoita vuoroille, vaikka normaalit tarkistukset eivät tätä sallisikaan.
- Käytettävyydestä tuli esiin toive siitä, että laboratorioiden aukioloajat näkyisivät sovelluksessa.
- Keplerin kalenteritietojen vieminen muihin järjestelmiin iCalendar muodossa olisi hyvä toteuttaa mahdollista tulevaa Korppi-integraatiota varten.
- Tietokannan alustusta tuotantoympäristöön varten tulisi tehdä skripti, joka ei syötä tietokantaan mallidataa.

6.5 Opiskelijan näkyisiin liittyvät jatkokehitysideat

Opiskelijan näkyisiin liittyen esitettiin projektin aikana seuraavat kehitysideat:

- Käyttäjän pitäisi pystyä poistumaan ryhmästä, jos kyseessä ei ole henkilökohtainen ryhmä.
- Varausta tehdessä olisi kalenterinäkylässä hyvä esittää myös valitun ryhmän muiden jäsenten menot.
- Jos käyttäjä ei voi poistaa ryhmää, ei hänelle pitäisi edes näyttää Poista ryhmä-painiketta.
- Uusi varaus-näkymään tulisi lisätä toiminto uuden ryhmän luomista tai vanhan ryhmän muokkaamista varten.
- Omien varauksien yhteydessä näytettävä Peru-painike pitäisi piilottaa tai ottaa pois käytöstä, jos vuoroa ei voi enää perua.
- Varausta ei saa perua, jos sen alkuun on aikaa vähemmän kuin 48 tuntia (tai jokin muu `config.py`:ssä säädetty aika). Tästä pitäisi ilmoittaa käyttäjälle joko uutta varausta tehdessä tai tulevia varauksia listatessa.
- Käyttäjärühmien yhteydessä oleva toisten käyttäjien haku toimii vain, jos hakusana täsmää täysin hakutulokseen. Tästä pitäisi kertoa käyttäjälle ja haku olisi syytä toteuttaa viisaammin.

- Kalenterinäkymissä pitäisi käyttäjälle ilmoittaa, jos hakuehtoja vastaavaa vapaata vuoroa ei löydyä (kuten listanäkymässä tehdään), ettei käyttäjä suotta hämmenny.
- Opiskelijoiden pitäisi kyetä näkemään ohjaajan yhteystiedot varauslistauksissa.

6.6 Ohjaajan näkymiin liittyvät jatkokehitysideat

Ohjaajan näkymiin liittyen esitettiin projektin aikana seuraavat kehitysideat:

- Ohjaajan tulisi kyetä näkemään myös muiden ohjaajien ohjausvuorot kalenterinäkyymässä (tämä toimii listanäkymässä).
- Ohjaajan pitäisi pystyä merkkamaan vuorolle ilmoittauneille tiloja sen mukaan, miten vuoro meni (esimerkiksi oli paikalla, työ jäi kesken tai ei tullut paikalle).
- Ohjaajien pitäisi kyetä vaihtamaan vuorojaan keskenään.

6.7 Pääkäyttäjän näkymiin liittyvät jatkokehitysideat

Pääkäyttäjän näkymiin liittyen esitettiin projektin aikana seuraavat kehitysideat:

- Pääkäyttäjän tulisi pystyä muokkaamaan olemassa olevia vuoroja.
- Pääkäyttäjän tulisi pystyä näkemään järjestelmän vuorot myös kalenterinäkyymässä (nykyisellään *Hallitse vuoroja*-näkyymän kalenterinäkyymässä näkyy pääkäyttäjän henkilökohtainen kalenteri).
- Pääkäyttäjän tulisi pystyä hoitamaan käyttäjien roolienhallinta käyttöliittymää käyttäen.
- Pääkäyttäjän tulisi pystyä lisäämään ilmoitustaulun ilmoituksille käännöksiä loogisella tavalla (nykyinen tapa kuvaillaan luvussa 6.1).
- Pääkäyttäjän tulisi pystyä muuttamaan ilmoitustaulun ilmoitusten järjestystä.

- Pääkäyttäjän tulisi pystyä määrittelemään ilmoitustaulun ilmoituksille kesto.
- Pääkäyttäjän tulisi pystyä tekemään ilmoituksesta "nosto", jolloin se näkyisi aina ensimmäisenä ilmoituksena.
- *Hallitse vuoroja*-näkyymässä näytettävien vuorojen aikarajoite ei toimi, joten se pitäisi joko toteuttaa tai piilottaa.
- *Hallitse vuoroja*-näkyymässä pitäisi pystyä erittelemään vuoroja tilojen perusteella (esimerkiksi suunnitteilla oleva, peruttu tai aktiivinen vuoro).
- *Kurssit*-näkyymässä pitäisi pystyä poistamaan työ kurssista.
- *Kurssit*- ja *Työt*-näkyymissä olisi syytä tarjota erilaisia hakutoimintoja (esimerkiksi "työt, joissa käytetään tiettyä resurssia" tai "kurssit, joilla tehdään tietty työ").

6.8 Tietojärjestelmän yleiskäyttöisyys

Tilaajan edustajien toive oli, että tietojärjestelmästä tehtäisiin yleiskäyttöinen, jotta sen saisi helposti myös muiden Jyväskylän yliopiston laitosten käyttöön. Tämä on otettu tietojärjestelmän suunnittelussa (etenkin tietokannan osalta) huomioon. Valitettavasti toteutuksessa sitä ei kuitenkaan ollut aikataulun puitteissa mahdollista ottaa juuri huomioon.

Tietojärjestelmästä saa nykyisellään tehtyä aika helposti toisen version samantyyppiseen käyttötarkoitukseen. Tällöin tarvitsee lähinnä muutella käyttöliittymän merkkijonoja, muokata käyttöliittymää vastaamaan omia tarpeitaan ja muokkaamaan palvelinpuolen toimintalogiikkaa vastaamaan omia tarpeitaan. Mikään näistä ei yksinään ole kovin vaikeaa, sillä tietojärjestelmän koodissa toistetaan pitkälti samoja kaavoja, joten esimerkkejä löytyy rutkasti. Tällaisesta muuntelusta suoriutuu nähdäkseni yksittäinenkin kehittäjä, jos hänellä on kokemusta HTML:stä, CSS:stä, JavaScriptistä ja Pythonista.

Mikäli on tarve lähteä kehittämään erityyppistä varausjärjestelmää, voi Kepler-järjestelmästä riisua kaikki itselleen tarpeettomat osat pois. Tällöin jäljelle jää varauspalvelun runko, joka jo noudattaa tiettyjä kaavoja ja säännönmukaisuuksia, tarjoaa valmiita apuluokkia ja tarjoaa esimerkki toteutusratkaisuja. Tämä voi auttaa

uuden järjestelmän kehittämisen aloittamisessa paljonkin, mutta se tuo mukanaan myös rajoituksia ja tekee helposti sokeaksi vaihtoehdoisille ratkaisuille.

Kepler-järjestelmästä voi myös yrittää tehdä yleispätevämmän. Suurimmat haasteet nykyisen toteutuksen yleiskäyttöisyydessä ovat fysiikan laitoksen vaatimukseen perustuvat tarkistukset REST-kutsuissa ja se, että käyttöliittymän koodi on tehty yksinomaan Kepleriä silmällä pitäen. Myös tietokannan rakenne voi muodostua ongelmaksi, jos järjestelmän käyttötarkoitus eroaa suuresti fysiikan laitoksen tarpeista (tietokantaan voi tutustua tarkemmin luvussa 5). Erilaiset toimintalogiikkaan liittyvät tarkistukset voisi hyvin eriyttää omiksi luokikseen (ja koodin luettavuuden nimissä näin kannattaisikin tehdä) ja käyttöliittymän koodikin on nykyisellään jo aika yleispätevää. Käyttöliittymän sisältö pitää luonnollisesti tehdä tai muokata erikseen eri käyttötapauksia varten. Näin ollen Kepler-järjestelmästä saisi jokseenkin helposti ja kohtalaisella työllä tehtyä yleispätevämmän version, mikäli ymmärtää Keplerin nykyisen toimintalogiikan hyvin.

7 Ohjeita ylläpitäjälle ja jatkokehittäjälle

Luvussa esitetään ohjeita sovelluksen ylläpitäjälle ja jatkokehittäjälle.

Pyramidin saa helpoiten asennettua pip:llä: `pip install pyramid`. Pyramid asentaa loput tarvitsemansa kirjastot automaattisesti tai ne ovat jo tulleet Kepler-järjestelmän tiedostojen mukana.

Ensimmäistä kertaa Pyramidia käytettäessä, pitää Kepler-järjestelmä asentaa. Tämä tapahtuu Kepler-järjestelmään juurikansiossa komennolla `python setup.py production` (kehittäessä tulee vaihtaa `production`-määre `development`-määreeksi).

Tietokannan saa alustettua käyttövalmiiksi ajamalla skriptin `initialize_Kepler_db.py`. Huomaa, että muuttamattomana tämä skripti laittaa tietokantaan paljon erilaista mallidataa.

Itse sovelluksen saa käynnistettyä komennolla `pserve production.ini`. Kepleriä kehittäessä kannattaa kuitenkin enemmän käyttää `pserve -reload development.ini`-komentoa.

Huomaa, että tuotantokäytössä on ehdottomasti käytettävä sitä vastaavaa määrettä `production` ja asetustiedostoa `production.ini`!

7.1 Ohjeita jatkokehittäjälle

Jos järjestelmää ajetaan `development.ini`:n asetuksilla, näkyy ylänavigointipalkin päällä joukko nimiä, joita klikkaamalla pääsee kirjautumaan sisään kyseisinä henkilöinä. Heistä Sulevi on pääkäyttäjä, Anna ja ? ovat ohjaajia ja Lisa ja Kalle ovat opiskelijoita. Tämän lisäksi ylänavigointipalkin päällä oikeassa reunassa on linkki *REST API*, josta pääsee tarkastelemaan REST rajapinnan dokumentaatiota HTML-muodossa (sieltä löytyy esimerkiksi kaikki Keplerin tukemat juliset kutsut, niiden kuvaus ja niiden vaatiman parametrit).

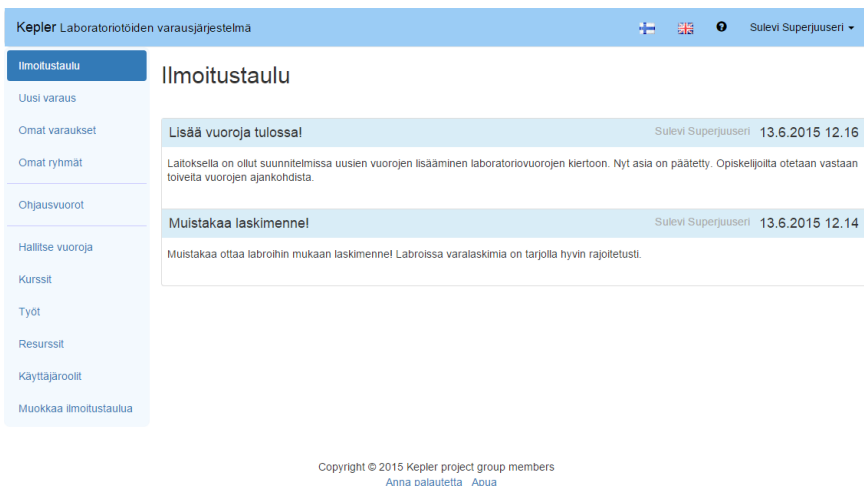
Lähes kaikki palvelimen tarjoaman rajapinnan ohjelmakoodi löytyy kansioista `/kepler/api/`. Lähes kaikki käyttöliittymäkoodi löytyy kansioista `/kepler/static`.

Kaikki järjestelmän eri näkymät löytyvä tiedostosta `/kepler/views.py`. Näille määritellään reitit `__init__.py` tiedostoissa, jotka sijaitsevat kansioissa `/kepler/__init__.py` ja `/kepler/api/__init__.py`.

Tietokannan `domain`-luokat määritellään omissa tiedostoissaan. Ne löytyvät kansioista `/kepler/models/`.

Käyttöliittymään liittyvä koodi löytyy kansioista `/kepler/static/`. HTML-sivujen luontiin käytetään Chameleon templateja.

Debuggauksessa kannattaa hyödyntää Pyramidin `debug-toolbaria`, jonka saa auki painamalla kuvassa 7.1 esitettyä painiketta.



Kepler Laboratoriotöiden varausjärjestelmä

Ilmoitustaulu

Lisää vuoroja tulossa! Sulevi Superjuuseri 13.6.2015 12.16

Laatoksella on ollut suunnitelmassa uusien vuorojen lisääminen laboratoriovuorojen kiertoon. Nyt asia on päätetty. Opiskelijoilta otetaan vastaan toiveita vuorojen ajankohdista.

Muistakaa laskimenne! Sulevi Superjuuseri 13.6.2015 12.14

Muistakaa ottaa labroihin mukaan laskimenne! Labroissa varalaskimia on tarjolla hyvin rajoitetusti.

Copyright © 2015 Kepler project group members
Anna palautetta Apua



Kuva 7.1: Pyramidin `debug-toolbar`-painike

Tarvittaessa voi testata palvelimen ominaisuuksia ilman, että kyseisiä ominaisuuksia varten on käyttöliittymää. Tämän voi tehdä selaimen kehittäjän työkaluilla ajamalla JavaScriptiä muodossa `Kepler.komento({json:dataa})`.

Huomaa, että REST API:n dokumentointi hoidetaan käyttäen `@rest_api` dekoratoria, ja kutsujen odottamat parametrit määritellään käyttäen `ApiParams`-luokkaa.

Jos tulee vastaan ongelmia OAuthin kanssa, kannattaa lähestyä sen toteuttanutta Atte Rätystä. Muut kehittäjät tuskin osaavat auttaa asiassa perehtymättä kyseiseen koodiin.

Tietokannasta ei saisi juuri koskaan poistaa mitään, vaan tieto pitää vain merkitä poistetuksi (tai ei-aktiiviseksi).

Tietokannan alustuksessa (`initialize_Kepler_db.py`) käytetään joitain aliohjelmiä, jotka eivät sovi käyttöön muualla järjestelmässä. Niiden tarkoituksena on ollut saada tietokantaan syötettyä helposti mallidataa, jotta järjestelmää voi testata.

Huomaathan, että moneen `Status-Type`-rakenteeseen on olemassa järjestelmässä määritellyjä vakioita (tiedostossa `/kepler/models/constants.py`). Hyödynnä näitä, niin välttyt turhilta virheiltä ja päänvaivoilta.

Sähköpostiviestit voi asettaa tallentumaan paikallisesti, niiden lähettämisen sijaan. Tämä tapahtuu laittamalla käytettyyn asetustiedostoon (eli jatkokehityksessä `development.ini`) `pyramid_includes=` määreen alle `pyramid_mailer.debug`. Näin sähköpostiominaisuuksien toimintaa voi testata paikallisesti. Nämä ”lähtevät” sähköpostit tallennetaan järjestelmän kansiorakenteen juureen, uuteen *mail*-kansioon.

7.2 Ohjeita ylläpitäjälle

`Production.ini`:stä löytyvät `[server:main]` asetukset tulee säätää vastaamaan käytössä olevaa palvelinta. `pyramid_mailer:in` asetukset tulee säätää `email_util.py`-tiedostossa (löytyy kansiopolusta `/kepler/api/`). Ohjeita `pyramid_mailer:in` käyttöön löytyy dokumentista ??.

Kepler-järjestelmä ei juuri koskaan poista tietokannasta mitään. Lähes kaikki vanhat tiedot ovat siten järjestelmässä tallessa, vaikka käyttäjät olisivatkin mielestään poistaneet ne.

Sovellusta asennettaessa on otettava huomioon seuraavat rajoitteet:

- Sovellus toimii testatusti käyttäen palvelinohjelmistona Apachea (versio ??) ja käyttöjärjestelmänä Red Hat Enterprise Linux (RHEL) 7.
- Sovellus toimii testatusti Python 3.4.3:lla.
- Sovellus ei toimi Python 2:lla.
- Sovellus ei toimi ilman JavaScriptiä.

Sovellusta käyttöönotettaessa on myös syytä ottaa huomioon seuraavat rajoitteet:

- Sovellus toimii testatusti Windows 7 ja Windows 8.1-käyttöjärjestelmissä.
- Sovellus toimii testatusti Ubuntu 15.04 -käyttöjärjestelmässä.
- Sovellus toimii testatusti selaimilla Google Chrome (versio ??), Chromium (versio 43.0.2357.81 (64-bit)), Mozilla Firefox (versio ??) ja Internet Explorer (versio ??).

8 Yhteenveto

Kepler-projekti toteutti Sovellusprojekti-kurssilla keväällä 2015 Jyväskylän yliopiston fysiikan laitokselle WWW-sovelluksena oppilaslaboratorioiden varausjärjestelmän. Sovelluksella oppilaat voivat varata vuoroja oppilaslaboratioihin, muodostaa keskenään ryhmiä ja nähdä järjestelmän esittämän ilmoitukset. Oppislaboriotöiden ohjaajat voivat nähdä tulevat ohjausvuoronsa ja niihin liittyvää tietoa. Sovelluksen pääkäyttäjät voivat luoda uusia ja hallinnoida vanhoja oppilaslaboratiovuoroja, oppilaslaboratiovuoroilla tehtäviä töitä, töihin liittyviä resursseja ja ilmoitustaulun ilmoituksia.

Toteutettu WWW-sovellus toimii pääosin asetettujen tavoitteiden mukaisesti, mutta joitakin asioita jäi projektin puitteissa toteuttamatta. Näistä merkittävimpiä ovat joistakin pääkäyttäjän toiminnallisuuksista puuttuvat tarkistukset. Lisäksi projektiryhmästä riippumattomista syistä suunniteltu integraatio opiskelijoiden Korppikalenterien kanssa jäi toteuttamatta. Jatkokehitysideoita esitettiin projektin aikana myös lukuisia.

Sovellus kehitettiin Pyramid-sovelluskehystä hyödyntäen. Sovelluksen toiminnan kannalta olennaisia komponentteja ovat myös Jyväskylän yliopiston tarjoama OAuth-rajapinta, SQLAlchemy, jQuery, Bootstrap ja FullCalendar. Sovelluksen merkittävimmät komponentit ovat WWW-pohjainen käyttöliittymä, palvelin ja tietokanta.

Lähteet

- [1] Joel Kivelä, Erkki Koskenkorva, Mika Lehtinen, Oskari Leppäaho ja Petri Partanen, "Liikkuva-sovellusprojekti, Sovellusraportti", saatavilla PDF-muodossa <URL: http://sovellusprojektit.it.jyu.fi/liikkuva/dokumentit/sovellusraportti/liikkuva_sovellusraportti_1.0.0.pdf>, Jyväskylän yliopisto, tietotekniikan laitos, 9.6.2014.
- [2] Joonas Konki, Anu Koskela, Mikko Kuhno, Henrik Paananen ja Atte Rätty, "Kepler-sovellusprojekti, Käytettävyytestaussuunnitelma", saatavilla PDF-muodossa <URL: http://sovellusprojektit.it.jyu.fi/kepler/testaus/kaytettavyystestaus/kepler_kaytettavyystestaussuunnitelma_1.0.0.pdf>, Jyväskylän yliopisto, tietotekniikan laitos, 2015.
- [3] Joonas Konki, Anu Koskela, Mikko Kuhno, Henrik Paananen ja Atte Rätty, "Kepler-sovellusprojekti, Käytettävyytestausmuistio", saatavilla PDF-muodossa <URL: http://sovellusprojektit.it.jyu.fi/kepler/testaus/kaytettavyystestaus/kepler_kaytettavyystestaus_muistio.pdf>, Jyväskylän yliopisto, tietotekniikan laitos, 2015.
- [4] Joonas Konki, Anu Koskela, Mikko Kuhno, Henrik Paananen ja Atte Rätty, "Kepler-sovellusprojekti, Järjestelmätestausraportti", saatavilla PDF-muodossa kahdessa osassa <URL: http://sovellusprojektit.it.jyu.fi/kepler/testaus/jarjestelmatestaus/Kepler_Jarjestelmatestaus_0.3.0_WinIE.pdf> ja <URL: http://sovellusprojektit.it.jyu.fi/kepler/testaus/jarjestelmatestaus/Kepler_Jarjestelmatestaus_0.3.0_LinuxFox.pdf>, Jyväskylän yliopisto, tietotekniikan laitos, 2015.
- [5] Joonas Konki, Anu Koskela, Mikko Kuhno, Henrik Paananen ja Atte Rätty, "Kepler-sovellusprojekti, Käyttöliittymän luokkadokumentaatio", saatavilla HTML-muodossa <URL: <http://sovellusprojektit.it.jyu.fi/kepler/application/>>

- class_documentation/javascript/>, Jyväskylän yliopisto, tietotekniikan laitos, 2015.
- [6] Joonas Konki, Anu Koskela, Mikko Kuhno, Henrik Paananen ja Atte Rätty, "Kepler-sovellusprojekti, Palvelimen luokkadokumentaatio", saatavilla HTML-muodossa <URL: http://sovellusprojektit.it.jyu.fi/kepler/application/class_documentation/python/>, Jyväskylän yliopisto, tietotekniikan laitos, 2015.
- [7] Joonas Konki, Anu Koskela, Mikko Kuhno, Henrik Paananen ja Atte Rätty, "Kepler-sovellusprojekti, Projektiraportti", saatavilla PDF-muodossa <URL: http://sovellusprojektit.it.jyu.fi/kepler/dokumentit/projektiraportti/kepler_projektiraportti_0.2.0.pdf>, Jyväskylän yliopisto, tietotekniikan laitos, 2015.
- [8] Joonas Konki, Anu Koskela, Mikko Kuhno, Henrik Paananen ja Atte Rätty, "Kepler-sovellusprojekti, Projektisuunnitelma", saatavilla PDF-muodossa <URL: http://sovellusprojektit.it.jyu.fi/kepler/dokumentit/projektisuunnitelma/kepler_projektisuunnitelma_1.0.0.pdf>, Jyväskylän yliopisto, tietotekniikan laitos, 2015.
- [9] Joonas Konki, Anu Koskela, Mikko Kuhno, Henrik Paananen ja Atte Rätty, "Kepler-sovellusprojekti, Vaatimusmäärittely", saatavilla PDF-muodossa <URL: http://sovellusprojektit.it.jyu.fi/kepler/dokumentit/vaatimusmaarittely/kepler_vaatimusmaarittely_0.8.0.pdf>, Jyväskylän yliopisto, tietotekniikan laitos, 2015.
- [10] "Jyväskylän yliopiston OAuth-rajapinnan ohjeet", saatavilla HTML-muodossa <URL: <https://dev-docs.app.jyu.fi/>>, Jyväskylän yliopisto, 2015
- [11] D. Hardt, Ed., "The OAuth 2.0 Authorization Framework", saatavilla HTML-muodossa <URL: <https://tools.ietf.org/html/rfc6749>>, Internet Engineering Task Force, 2012
- [12] Ben Bangert ja Mike Bayer, "Beaker documentatiom", saatavilla HTML-muodossa <URL: <https://beaker.readthedocs.org/en/latest/>>, 2012

- [13] Bootstrap core team, "Bootstrap: The world's most popular mobile-first and responsive front-end framework.", saatavilla HTML-muodossa <URL: <http://getbootstrap.com/>>, 2015
- [14] "dateutil - powerful extensions to datetime" saatavilla HTML-muodossa <URL: <https://dateutil.readthedocs.org/en/latest/>>, 2015
- [15] "Docutils: Documentation Utilities" saatavilla HTML-muodossa <URL: <http://docutils.sourceforge.net/>>, 2014
- [16] "A JavaScript event calendar. Customizable and open source.", saatavilla HTML-muodossa <URL: <http://docutils.sourceforge.net/>>, Adam Shaw, 2015
- [17] "jQuery: write less, do more", saatavilla HTML-muodossa <URL: <https://jquery.com/>>, jQuery Foundation, 2015
- [18] "iCalendar package", saatavilla HTML-muodossa <URL: <https://icalendar.readthedocs.org/en/latest/usage.html>>, Plone Foundation, 2015
- [19] Agendaless Consulting, "The Pyramid Web Framework", saatavilla HTML-muodossa <URL: <http://docs.pylonsproject.org/projects/pyramid/en/latest/>>, Pylons Project, 2015
- [20] Agendaless Consulting, "pyramid_beaker", saatavilla HTML-muodossa <URL: <https://pylons.readthedocs.org/projects/pyramid-beaker/en/latest/>>, Pylons Project, 2012
- [21] Dan Jacob, "pyramid_mailer", saatavilla HTML-muodossa <URL: <http://pylons.readthedocs.org/projects/pyramid-mailer/en/latest/>>, Pylons Project, 2015
- [22] "Requests: HTTP for Humans", saatavilla HTML-muodossa <URL: <http://docs.python-requests.org/en/latest/>>, Kenneth Reitz Project, 2015
- [23] "The Python SQL Toolkit and Object Relational Mapper", saatavilla HTML-muodossa <URL: <http://www.sqlalchemy.org/>>, SQL Alchemy authors and contributors, 2015