

Tallinna Ülikool
Informaatika Instituut

Arvutiklassi broneerimise veebirakendus

Eesti koolidele

Bakalaureusetöö

Autor: Raimo Virolainen

Juhendaja: Mart Laanpere

Autor: „2014

Juhendaja: „2014

Instituudi direktor: „2014

Tallinn 2014

Autorideklaratsioon

Kinnitan, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

(kuupäev)

.....

(allkiri)

Sisukord

Sissejuhatus	5
1 Olemasolevad lahendused	7
1.1 Doodle.....	7
1.2 Classroombookings.....	8
1.3 Asio EduERP Classroom Booking System	10
1.4 Multi-View Calendar for Joomla.....	11
2 Rakenduse disain	13
2.1 Esialgne disain	13
2.2 Disainisessioon	14
3 Loodud rakendus	17
3.1 Tehnoloogiad	17
3.2 Nõuded.....	19
3.2.1 Mittefunktsionaalsed nõuded.....	19
3.2.2 Funktsionaalsed nõuded	20
3.3 Andmemudel.....	21
3.4 Realiseerimine	23
3.4.1 Kasutajate registreerimine ja haldamine.....	24
3.4.2 Broneeringu tegemine.....	27
3.4.3 Konflikti lahendamine	29
4 Evalvatsioon	32
5 Edasiarendamise võimalused.....	33
Kokkuvõte	34
Summary: Computer classroom scheduling web application for Estonian schools.....	35
Kasutatud kirjandus	36

LISA	38
Lisa 1. Persoona 1: Peeter.....	39
Lisa 2. Persoona 2: Eve	40
Lisa 3. Persoona 3: Linda	41
Lisa 4. Persoona 4: Juhan	42
Lisa 5. Stsenaarium 1: Kooli registreerimine	43
Lisa 6. Stsenaarium 2: Ühekordne broneering	43
Lisa 7. Stsenaarium 3: Pika-ajaline broneering	44
Lisa 8. Stsenaarium 4: Konflikti lahendamine	44
Lisa 9. Stsenaarium 5: Statistika	45
Lisa 10. Kasutusjuhend	45

Sissejuhatus

Käesoleva bakalaureuse töö eesmärgiks on luua veebirakendus, mis võimaldab Eesti koolide õpetajatel broneerida arvutiklasse. Hetkel selline ühtne süsteem Eestis puudub ja arvutiklasside broneerimine koolides toimub aegunud meetoditel. Loodava rakenduse eesmärgiks on koguda andmeid eesti koolide arvutiklasside kasutamisest. Andmeid kogutakse teaduslikel eesmärkidel. Näiteks oleks kogutud andmete põhjal võimalik tõestada hüpotees, et eesti koolides arvutiklasside vajadus väheneb, seoses isiklike arvutite kasutamise tõusuga.

Eesti koolides kasutatakse arvutiklasside broneerimisel paremal juhusel mõnda olemasolevat tarkvara ja halvimal juhusel tehakse seda paber kandjal. Samuti tihti peale broneeringute tegemiseks kasutatav tarkvara ei täida broneerimise ülesannet kõige paremini ehk siis see tähendab, et rakendusena kasutatakse mõnd väga üldist tarkvara, mis küll aitab broneerimise ülesannet täita, kuid pole rakenduse loomise ja kasutamise põhieesmärk.

Teema pakkus välja Haridustehnoloogia Keskuse vanemteadur Mart Laanpere, kes on ka käesoleva töö juhendajaks. Autor valis teema isiklikust huvist veebirakenduste arendamise vastu ning vajadusest luua broneerimisrakendus kasutamiseks muus valdkonnas.

Käesolevas töös loodava rakenduse eesmärgiks on realiseerida broneerimine võimalikult lihtsalt. Rakenduse disain loodi disainipõhise uuringu meetodil, mille käigus selgitati välja rakenduse funktsionaalsused ja mille tulemuseks on prototüüp.

Rakendus luuakse kasutades Joomla sisuhaldussüsteemi ja Joomla pakutavat Fabrik komponenti. Rakenduse loomisel kasutatakse Joomla versiooni 3.2.2 ja Fabrik'u versiooni 3.1.

Esimeses peatükis teeb autor ülevaate olemasolevates broneerimisrakendustest, mis autori arvates kõige rohkem sarnanevad loodava rakendusega. Teises peatükis kirjeldatakse disainiprotsessi. Kolmandas peatükis antakse ülevaade loodud rakendusest. Tuuakse näiteid

funktsionaalsuste realiseerimisest rakenduses. Neljandas peatükis antakse hinnang loodud rakendusele ja viiendas pakutakse välja võimalusi rakenduse edasiarendamiseks.

1 Olemasolevad lahendused

Veebist on võimalik leida väga palju erinevaid lahendusi broneeringute tegemiseks. Antud peatükis on toodud välja neli lahendust, mis funktsionaalsuste ja eesmärkidega kõige enam loodava rakendusega sarnanevad. Iga lahenduse juures toon välja eelised ja puudused, mis nendel rakendustel esinevad käesolevas töös loodava rakenduse vajadustest lähtudes.

1.1 Doodle

Doodle on veebirakendus, mille abil kasutajad saavad kiiresti ja lihtsalt leida ühiselt sobiva aja mingi sündmuse jaoks. Doodle'it saaks rakendada ka lihtsa broneeringusüsteemi tegemiseks, kuid käesolevas töös nõutud veebirakenduse tegemiseks jääb puudu funktsionaalsustest. Doodle on tasuta, kuid pakutakse ka tasulisi versiooni, millega funktsionaalsusi juurde lisatakse. [7]

Doodle'i põhilisteks funktsionaalsusteks on:

- Sündmuse loomine – Doodle'is on võimalik luua sündmusi ja määrata sündmuse võimalikud toimumise ajad.
- Sündmuse tabel – sündmuse loomisel genereerib Doodle võimalike toimumise aegade tabeli, mis on küsitluse vormis. Küsitlusest osavõtjad saavad märkida, millistel aegadel saavad sündmusest osa võtta.
- Sündmuse administreerimine – sündmuse looja saab sündmust administreerida. Sündmust saab ka eksportida Excelisse ja PDF-i.
- Sündmuste kommenteerimine
- Muudatustest teavitamine – sündmuse loojat teavitatakse e-maili teel uute kommentaaride lisamisest ja küsitlusest osavõtmistest.

- Osavõtjate kutsumine – e-maili teel saab kutsuda inimesi küsitlusest osavõtma. See funktsionaalsus on sisselogitud kasutajatel.
- Kalendri vaade – küsitlust saab näha kalendri vaates. See funktsionaalsus on sisselogitud kasutajatel.

Doodle'i eelised:

- Doodle'i võlu on selles, et ta on väga lihtsasti kasutatav.
- Doodle'i kasutamiseks ei pea looma kasutajakontot. Lisafunktsionaalsuste lisamiseks tuleb kasutajakonto luua või autentida läbi Gmail'i või Facebook'i.
- Doodle on tasuta.

Doodle'i puudused:

- Doodle'ile sündmusele ei saa lisada välju. Klassiruumide broneerimisel ei saaks lisada klasse.
- Doodle'iga ei saaks koguda andmeid kooli tasandi statistika tegemiseks.
- Doodle'il ei ole eesti keelset tõlget.

Käesolevas töös loodav rakendus peaks olema sama lihtsasti kasutatav kui Doodle, kuid teistsuguse funktsionaalsusega.

1.2 Classroombookings

Classroombookings on 2006. aastal Craig A. Rodway poolt loodud avatud lähtekoodiga veebirakendus, mis on mõeldud klasside broneerimiseks. Rakendus on loodud „Codeigniter“ raamistikus ja rakendusest on väljas esimene versioon. Rakendus on veidi aegunud, kuid arenduses on teine versioon rakendusest. Rakendus sarnaneb Doodle'iga, kuid täidab ka käesolevas töös nõutavaid funktsionaalsusi. Rakenduses on võimalik ära määrata akadeemiline periood, tundide toimumise ajad ja klassid, kuhu on võimalik broneeringuid teha. Rakendus võimaldab ka eemaldada broneerimise konfliktid, lubades ainult ühel kasutajal üheks tunniajaks klassi broneerida. [3]

Classroombookings rakenduse funktsionaalsused:

- Kooli detailid – kooli kohta käiv informatsioon (kooli nimi, veebileht, logo jne). Saab ka määrata „*Booking in advance*“ valikut, millega määratakse mitu päeva ette saavad õpetajad broneeringut teha.
- Tunniaegade määramine – saab määrata tundide toimumise ajad.
- Klasside määramine – saab määrata koolis olevad klassid, mida rakenduses broneerida on võimalik.
- Akadeemilise perioodi määramine
- Ühekordne broneering – õpetaja rollis kasutajad saavad teha ühekordseid broneeringuid.
- Pikaajaline broneering – administraatori rollis kasutajad saavad teha ka pikaajalisi broneeringuid.

Classroombookings rakendus tundub olevat veel arengujärgus ja ei pruugi olla kõige usaldusväärsem. Näiteks esinesid rakenduse kasutamisel mõned vead. Rakendus võimaldab broneeringuid teha ainult ühes koolis. Kui seda Eesti koolides kasutada, peaks iga kool selle rakenduse endale veebiserverisse installeerima. Sellisel juhul kahjuks ei õnnestuks koguda andmeid Eesti koolide arvutiklasside kasutuse kohta, mis on käesoleva töö üheks eesmärgiks.

Classroombookings eelised:

- Doodle'ile sarnane broneerimine. Väga lihtsasti kasutatav.
- Paljud funktsionaalsused on realiseeritud, mis nõutavad käesolevas töös loodavas rakenduses.
- Avatud lähtekoodiga ja tasuta.

Classroombookings puudused:

- Rakendus pole usaldusväärne. Esinevad mõned vead.
- Rakendus on mõeldud klasside broneerimiseks ühes koolis.
- Andmete kogumine statistika eesmärgil ei ole võimalik.
- Rakendus on inglise keelne.

1.3 Asio EduERP Classroom Booking System

Asio klassi broneerimise süsteem (edaspidi Asio) on veebirakendus, mis on mõeldud klassiruumide, õpetajate, kursuste ja seadmete haldamiseks ja broneerimiseks ning on eelkõige mõeldud kasutamiseks ülikoolides ja teistes suuremates asutustes. Asio on kasutusel paljudes koolides (enamasti Soomes) ja on põhjalikult testitud ja usaldusväärne rakendus. Asio on kasutusel ka Tallinna Ülikoolis. Asio on loodud kasutades PHP-d ja MySQL-i. Asio puhul on tegemist tasulise tarkvaraga. [2]

Rakenduses on kolm juurdepääsu õiguste taset (administraator, õpetajad, õpilased), mis määravad, milline kasutajaliides esitatakse ja millised funktsionaalsused kasutajal on.

Asio klassi broneerimise süsteemi põhilised funktsionaalsused:

- Broneerimise kalendrid ruumide, kursuste, seadmete ja õpetajate kohta.
- Erinevad broneerimise tüübid – võimaldab ühekordseid, korduvaid, mitmekordseid (mitu ruumi, inimest või seadet ühes broneeringus) ja grupi broneeringuid.
- Rakenduses on võimalik lisada hooneid ja määrata klassid erinevates hoonetes.
- Kinnitatud ja kinnitamata omadusega broneerigu tegemised.
- Vaba aja ja seadme otsing.
- Erineva staatusega broneeringud kalendris erinevate värvidega.

Täpsema ülevaate funktsionaalsustest saab Asio dokumentatsioonist. [1]

Asio täidab enamasti kõik käesolevas töös loodava rakenduse funktsionaalsused. Sarnaselt Classroombookings rakendusele on Asio ühe kooli keskne ning seega mitme kooli broneeringuandmete kogumine ei ole võimalik. Kuna Asio on tasuline ning eelkõige mõeldud ülikoolidele, siis Eesti väiksemates koolides ASIO-t kasutada ei oleks kõige efektiivsem ja on liialt kulukas.

ASIO eelised:

- Usaldusväärne – Asio on põhjalikult testitud ja on kasutusel paljudes koolides.
- Funktsionaalsuste poolest kõige võimekam peatükis väljatoodud lahendustest.
- Paindlik – Asio rakendust on võimalik kohandada vastavalt nõuetele.

Asio puudused:

- Asio on ühe kooli keskne. Erinevate koolide broneerimise andmeid pole võimalik koguda.
- Asio puhul on tegemist tasulise lahendusega.
- Lihtsa klassi broneerimise tegemiseks on Asio liiga keeruline ja liialt võimekas. Ei oleks rakenduse efektiivne kasutamine.
- Kuigi autor Asio täpset hinda ei tea, siis suure tõenäosusega oleks Asio ka liiga kulukas, arvestades rakenduse võimekust.

1.4 Multi-View Calendar for Joomla

Käesoleva töös loodav rakendus luuakse kasutades Joomla raamistikku. Joomlaile võib leida palju erinevaid laiendusi, mis broneerimise funktsionaalsust realiseerivad. „Multi-View Calendar“ on Joomla komponent, mis autori arvates peaks kõige enam sarnanema töös loodavale rakendusele. „Multi-View Calendar“ komponent on kalender, millel on erinevad vaated (kuu, nädala, päeva vaated). Kasutajad saavad kalendrist uusi sündmusi lisada ja määrata sündmustele erinevaid omadusi (värv, toimumise aeg, asukoht jne). Võimalik on lisada ka korduvaid üritusi, mis määratud intervalli järel korduvad ning määrata, millistel kasutajatel või kasutajagruppidel on õigus sündmusi lisada, muuta ja kustutada. Seda komponenti on väga lihtne kasutada, kuid tasuta versioonil on sündmuste kogusele kehtestatud piirang. [8]

„Multi-View Calendar“ funktsionaalsused:

- Erinevad vaated kalendris (päeva, nädala, kuu, mitme kuu).
- Konfigureeritav administratsiooniliidesest.
- Kasutajatele saab võimaldada ürituste muutmise kasutajaliidesest.
- Rakenduses on kalendriks erinevad värvi skeemid.
- Kasutajaõiguste haldamine
- Pikajalise broneeringu võimalus.

„Multi-View Calendar“ eelised:

- Lihtne kasutada.
- Komponent on tasuta (olemas ka tasuline versioon).

„Multi-View Calendar“ puudused:

- Tasuta komponendi versioonile on kehtestatud piirangud. Kalendrisse saab lisada kõigest viis broneeringut.
- Kõik käesolevas töös loodava rakenduse nõutavad funktsionaalsused ei ole realiseeritud. Näiteks oleks vaja lisada klasside ja tunniaegade valiku võimalus. Samuti tuleks lahendada broneeringu konflikt.

2 Rakenduse disain

Käesoleva töö uurimus viidi läbi disainipõhise uuringu meetodil (ingl k. *Design-based research*). Rakenduse loomisel on võetud eelduseks asjaolu, et Eesti koolides arvutiklasside broneerimine toimub aegunud meetoditel, mis põhineb juhendajalt saadud informatsioonilt. Disainiprotsessis kasutatakse stsenaariumipõhist meetodit. Stsenaariumipõhise meetodi eesmärk ei ole defineerida funktsionaalsuste spetsifikatsiooni, vaid kirjeldada, kuidas kasutajad rakendust kasutades mingi eesmärgi saavutavad. Stsenaariumipõhise meetodi korral kirjeldatakse varajases arenguprotsessis loodava rakenduse kasutamist.[9]

Stsenaariumid on lood, mis kirjeldavad sündmuste ja tegevuste jada, mis eesmärgini viivad. Stsenaariumite eesmärgiks on tekitada arutelu disainivigade väljaselgitamiseks.[9]

Tulevase rakenduse kasutajate kohta tuleks kirjutada persoonad. Personad on kasutajate kirjeldused, kes stsenaariumites osalevad. Igal stsenaariumil on üks või enam persoonat.

Disainisessiooni eesmärgiks on persoonadega kokkusobivate isikutega läbi arutada stsenaariumid, et leida disainis vigu ja puudujääke. Tulemuseks peaks tekkima visualiseering rakendusest, mille põhjal prototüüpi saaks luua.

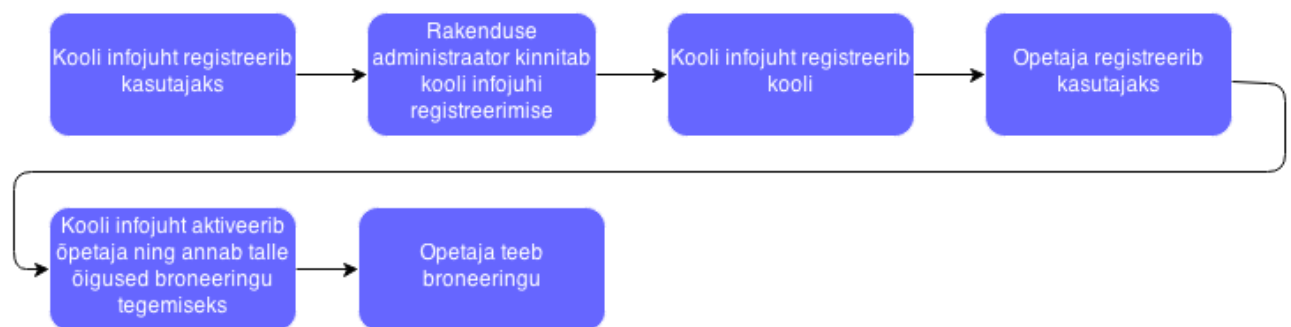
2.1 Esialgne disain

Disainimisel võeti aluseks rakenduse põhieesmärgid. Eesmärgiks on luua veebirakendus, mis võimaldab kasutajatel broneerida arvutiklasse.

Rakenduse disainimine sai alguse pannes kirja persoonad (Vt Lisa 1-4) ja stsenaariumid (Vt Lisa 5-9). Selle tulemusena selgus, et rakendusel on neli kasutajatüüpi: kooli infojuht, direktor, informaatikaõpetaja ja õpetaja.

Kasutajatüüpide põhiomadused:

- Kooli infojuht – registreerib kooli rakenduses ning haldab vastava kooli kasutajaid.
- Direktor – näeb rakenduses statistikat klasside kasutamise kohta.
- Informaatikaõpetaja – saab broneerida klasse ja lisaks võimalus broneerida klasse pikemaks perioodiks. Samuti prioriteet klassi broneerimisel.
- Õpetaja – saab teha ühekordset klassi broneerimist.



Joonis 1 – Rakenduse kasutamise süžeetahvel

Stsenaariumid kirjeldasid täpsemalt enamasti kahte protsessi – kooli registreerimine ning broneeringu tegemine. Kooli registreerimisel määrab kooli infojuht milliseid klasse, millistel aegadel ja milliseks perioodiks õpetajad broneerida saavad. Esialgse disaini kohaselt saab õpetaja rolli määratud kasutaja broneerida klassi ühekordselt ning informaatikaõpetaja rolli määratud kasutaja saab broneerida klassi terveks perioodiks. Juhul kui tekib konflikt ehk kui ühte klassi soovib samaks ajaks broneerida kaks erinevat õpetajat, antakse prioriteet informaatikaõpetajale. Eelnev broneering kustutatakse ja broneeringu teinud õpetajat teavitatakse broneeringu tühistamisest e-maili teel.

2.2 Disainisessioon

25. aprillil 2014 aastal toimus disainisessioon, mille käigus arutati läbi stsenaariumid ja kuna disainisessiooni toimumise ajaks oli valmis ka algne prototüüp esialgse disainiga rakendusest, siis mängiti ka rakenduse peal stsenaariumid läbi. Disainisessioon toimus autori juhendamisel koos kahe eksperdiga, kellest esimene oli ülikooli teadur ja teine kooli infojuht ühes Eesti koolis.

Disainisessioon algas kooli registreerimise stsenaariumi katsetamisega. Mõlemad eksperdid registreerisid ennast rakenduses kasutajaks valides registreerides sama kooli. Autor, kes mängis rakenduse administraatori rolli, määras ühele kasutajale kooli infojuhi rolli. Seejärel sai infojuhi rolli määratud kasutaja rakendusse sisselogida ja sai alustada kooli registreerimist. Kooli registreerimine õnnestus ja stsenaarium oli katsetatud. Selgusid järgnevad kujundusega seotud vead:

- Kasutajaks registreerimisel valiti juba kool, seega kooli registreerimisel on kasutaja juba seotud kooliga ja ei peaks saama enam kooli valida.
- Kooli registreerimise vorm on jaotatud kolmeks grupiks, mis eraldi lehtedel kuvatakse. Need grupid on: perioodi määramine, tunniaegade määramine ja klasside määramine. Salvesta nupp ilmub kõigil kolmel leheküljel, kuid on aktiivne kõigest viimasel. Inaktiivset salvestusnuppu ei tohiks kuvada.
- Tunniaegade määramisel oleks parem, kui on kuvatud ka tunni järjekorra number.

Järgmisena katsetati klassi broneeringu tegemist. Kooli infojuhi rolli määratud kasutaja läks rakenduses kasutajate halduse lehele ning määras teisele kasutajale õpetaja rolli. Seejärel tehti broneering, kus õpetaja täitis broneeringu vormi väljad. Väljadeks olid: õppeaine, broneeringu nimi, aeg, kuupäev, klass ja märkused. Selgus, et broneeringu nimi on üleliigne. Vaikimisi võiks broneeringu nimeks olla õppeaine nimi ja väli võiks olla peidetud, kuid kui valida õppeaine menüüst „muu“, siis kuvatakse väli ning saab täpsustada broneeringu nime. Klassi broneerimise vaates, kus kuvatakse kalender, on võimalik ka broneeringuid kalendris filtreerida klasside kaupa. Selle kuvatakse rippmenüü, kus kuvatakse klassid (olemasolevate broneeringute kohta). Kujunduslikult selle rippmenüü asukoht ei tohiks olla kalendri kohal, sest esimese asjana peaks paistma lisamise nupp.

Katsetati ka pikaajalist broneeringut muutes kasutaja kasutajagruppi ehk kasutajagrupiks määrati informaatikaõpetaja. Klassi broneerimise vormis tekkis lisa väli, mis võimaldas pikaajalist broneeringut sooritada ja pakkudes välja perioodi, mille vältel broneering tehakse. Pikaajalise broneeringu tegemine ebaõnnestus, kuna hiljem selgus, et serverist oli puudu vastav PHP skript. Katsetamise hetkel ei olnud kool infojuhi rolli määratud kasutajal õigusi klassi broneerida, kuigi temal peaks need õigused olema.

Arutati ka läbi konflikti lahendamise stsenaarium, mille käigus selgus, et automaatne konflikti lahendamine informaatikaõpetaja kasuks ei ole kõige arukam. Samuti peaks olema võimalik kahel tavalisel õpetajal sama klassi üheks ja samaks ajaks broneerida. Parem lahendus oleks konflikti probleem lahendada läbirääkimiste teel. Kui õpetaja soovib broneerida klassi juba olemasoleva broneeringuga ajaks, siis uus broneering küll tehakse, kuid määratakse nii uuele kui ka varasemale broneeringule kinnitamata staatus. Olemasoleva broneeringu teinud õpetajale saadetakse e-mail kahe lingiga, millega saab otsustada, kas loovutab oma aja ja teeb broneeringu uuele ajale või keeldub aega loovutamast. Sel juhul viimasena broneeringu teinud õpetaja broneering tühistatakse ning teda emaili teel teavitatakse. Informaatikaõpetaja roll rakenduses muutub ebaoluliseks. Ainult pikaajalise broneeringu keelamiseks pole rolli vaja, sest reaalsuses puudub tavaõpetajal vajadus pikaajaliseks broneeringuks.

Lühidalt arutati läbi ka statistika kuvamise stsenaarium. Katsetamise hetkeks seda valmis ei olnud, küll aga võimaldab rakenduses loomisel kasutatav Fabrik komponent kõiki list'ide andmeid CSV faili eksportida ning selle funktsionaalsuse lisamine ei ole prioriteetne. Samuti oleks hea, kui oleks võimalik kalendri nädalavaadet printida, luues sellest näiteks PDF fail.

Lühike kokkuvõtte disainisessioonist:

- Kooli valimise eemaldamine kooli registreerimisel. Kasutaja registreerimisel seotakse kasutaja kooliga.
- Konflikti lahendamise ümbertegemine. Konflikt lahendatakse läbiräägimiste teel.
- Informaatikaõpetaja rolli kaotamine. Pikaajalisi broneeringuid võivad teha ka õpetajad (kuid üldjuhul puudub selleks vajadus).
- Broneeringu nime vaikimisi väärtus on õppeaine nimi.
- Kalendris nädalavaate printimise funktsionaalsuse lisamine (näiteks PDF formaat).
- Kujunduslikud muudatused.

3 Loodud rakendus

3.1 Tehnoloogiad

Rakendus on loodud kasutades Joomla raamistikku. Joomla on tasuta kättesaadav ning avatud lähtekoodiga raamistik ja sisuhaldussüsteem (ingl. k *CMS – Content Management System*) – välja antud GPL litsentsi alusel. Joomla on üks populaarsematest sisuhaldussüsteemidest, mille tugevateks külgedeks on näiteks lihtne kasutatavus ja installeerimine ning laiendatavus. Joomla baasinstalleerimisega lisatakse juba palju erinevaid funktsionaalsusi, kuid läbi laienduste on võimalik neid veel juurde lisada. Joomla laiendused jagunevad viieks, millest igaüks käsitleb spetsiifilist funktsionaalsust: [5]

- Komponent (*Component*) – üldjuhul on komponendid kõige mahukamad ja keerukamad laienduste liikidest. Joomla komponente võib käsitleda kui eraldiseisvaid rakendusi või süsteemi koostisosi, mis mingit spetsiifilist funktsionaalsust lisavad. Enamustel komponentidel on kaks osa: veebisaidi (*front-end*) osa ja administraatori (*back-end*) osa. Näiteks „Content“(*com_content*) komponent, mis tegeleb sisu esitamisega kasutajatele veebilehel veebisaidi osas ja sisu muutmise veebilehe administraatori osas.
- Moodul (*Module*) – moodulid on väiksemamahulised ja rohkem paindlikumad laiendused, mida kasutatakse lehekülgede esitamisel. Mooduleid kasutatakse komponentide lisadena, kuid võivad olla ka eraldiseisvad. Näiteks Login moodul, mis esitab sisselogimise vormi (sisselogimise protsessiga tegeleb *com_users* komponent).
- Plugin (*Plugin*) – plugin on laiendus, mis töötleb sündmusi. Kui veebilehel või veebirakenduses mingi sündmus või protsess käivitatakse, siis käivitatakse ka plugin, mis rakenduses on määratud antud sündmust või protsessi töötleva.
- Disainimall (*Template*) – disainimall on Joomla veebilehe kujundus. Sellega saab muuta veebilehe välimust.

- Keel (*Language*) – Joomla! saab lisada erinevaid tõlkeid. Keelte abil saab koostada mitmekeelseid veebilehti.

Rakendus on loodud kasutades Fabrik komponenti. Fabrik on samamoodi avatud lähtekoodiga ja tasuta. Tasu eest saab tellida kasutajatuge. Fabrik on Joomla! komponent, mis võimaldab paindlike veebirakenduste loomist. Fabrik võimaldab koguda vormide kaudu andmeid kasutajatelt, salvestada saadud andmeid MySQL andmebaasi ja esitada neid andmeid erinevatel viisidel. Lihtsamaid veebirakendusi saab luua ilma koodi kirjutamata ning seega puudub otsene programmeerimisoskuse vajadus. Käesolevas töös loodud veebirakendus vajab spetsiifiliste funktsionaalsuste realiseerimisel lisakoodi kirjutamist.

Fabrik komponendi põhiosad: [4]

- Listid (*Lists*) – list'id on mõeldud vormide poolt kogutud või olemasolevate andmebaasi tabelite andmete esitamiseks. List'e võib vaadelda kui andmetabelite vaateid. Näiteks list'idega on võimalik esitada andmeid mitmest erinevast andmetabelist, erinevad list'id võivad olla seotud sama andmetabeliga ning igale list'ile saab lisada ligipääsuõigusi ja andmete filtreerimist.
- Vormid (*Forms*) – vormide ülesandeks on koguda andmeid kasutajate käest.
- Grupid (*Groups*) – grupid on elementide kogumid vormis. Gruppe või vaadelda kui „*<fieldset>*“ elementi HTML-is. Igas vormis võib olla mitmeid gruppe ning elementidele peab olema määratud grupp selleks, et neid vormis esitada.
- Elemendid (*Elements*) – elemendid on andmete väljad vormis ning samuti veerud andmetabelis. Elementidele määratud nime ja tüübi järgi määratakse ka vastava veeru nimi ja tüüp andmetabelis.
- Visualiseerimine (*Visualizations*) – visualiseerimise vahendi abil saab andmeid andmebaasist esitada läbi visuaalsete lahenduste. Näiteks on visualiseerimise väljundid kalender, kaart ja diagramm.
- Pakett (*Package*) – pkett vahendi abil on võimalik luua installeeritavad komponendid list'ide, vormide ja visualiseerimiste kogumikest. Pakette on võimalik installeerida Joomla! veebilehtedele juhul kui nendele on Fabrik komponent installeeritud.
- Ühendused (*Connections*) – ühenduse abil on võimalik luua ühendus andmebaasiga.

- Ajakava (*Schedule*) – ajakava vahendiga on võimalik käivitada ülesandeid määratud intervallide järel.

Tehnoloogiate valiku tegi autor. Esialgsete nõuete väljaselgitamisel selgus, et teatud funktsionaalsused saab lisatud, kui luua rakendus mõne olemasoleva raamistiku peale. Lisaks eelpool mainitud Joomla laienduste võimaluste lisamisele on Joomla juba kaasas kasutajate haldamise funktsionaalsus. Joomla kasutamisel lisatakse rakendusele juba kasutajate registreerimise, autentimise ja kasutajagruppide ning nende õiguste funktsionaalsused. Olles eelnevalt veebiarenduses Joomla kasutanud, osutuski Joomla kasutamine rakenduse loomisel valikuks.

Joomla laienduste hulgast võib leida palju erinevaid broneerimiseks mõeldud nii tasuta kui ka tasulisi rakendusi. Põhjuseks, miks enamus nendest lahendustest ei sobi käesolevas töös loodava rakenduse loomiseks on asjaolu, et neid on raske või isegi võimatu vajadustele kohandada. Näiteks lisafunktsionaalsuste lisamine. Kasutades Fabrik komponenti saab luua võimekaid veebirakendusi, antud juhul luuakse broneerimise rakendus. Fabrik komponendi kasutamise valikul sai määravaks asjaolu, et rakenduse loojal on täielik kontroll rakenduse toimimise üle.

3.2 Nõuded

Rakenduse loomisel on üheks etapiks ka nõuete analüüs. Nõuded tulenevad rakenduse põhieesmärkidest ja rakenduse disainiprotsessis väljaselgitatud järeldustest. Nõuded jagunevad funktsionaalseteks ja mittefunktsionaalseteks. Käesolevas peatükis esitatakse ülevaade loodava rakenduse nõuetest.

3.2.1 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded antud rakendusele on järgmised:

- Käesolevas töös loodav rakendus on veebirakendus, seega peab töötama vähemalt kahes neljast levinud veebilehitsejast (Firefox, Google Chrome, IE, Safari). Rakendus peab töötama vähemalt viimase stabiilse versiooniga konkreetsest veebilehitsejast.

- Rakendus on loodud kasutades Joomla versiooni 3.2.2. Sellest tulenevalt süsteeminõuded. (Vt Tabel 1)[6]

Tabel 1 – Mittefunktsionaalsed nõuded – Joomla versiooni 3.x tehnilised nõuded

Tarkvara	Soovitatud	Miinum
PHP	5.4 +	5.3.10 +
Toetatud andmebaasid:		
MySQL	5.1 +	5.1 +
MSSQL	10.50.1600.1 +	10.50.1600.1 +
PostgreSQL	8.3.18 +	8.3.18 +
Toetatud veebiserverid:		
Apache	2.x +	2.x +
Nginx	1.1	1.0
Microsoft IIS	7	7

3.2.2 Funktsionaalsed nõuded

Esialgse rakenduse disaini kohaselt oli rakendusel neli peamist kasutajatüüpi. Pärast disainisessiooni selgus, et piisab kolmest: kooli infojuht, õpetajad ja direktor.

1. Registreerimine – rakendust soovivad kasutajad registreerivad end kasutajateks. Registreerimisel valitakse, millise kooli nimekirja ennast registreeritakse. Enne rakenduse kasutamist peab kooli infojuht kasutajad aktiveerima ja kasutajatele kasutajagrupperi määrama.
2. Kasutaja andmete muutmine – kasutajad peavad saama registreerimisel sisestatud andmeid muuta (nimi, parool, e-mail jne).
3. Kooli registreerimine (perioodide, tunniaegade ja klasside määramine) – kooli saab registreerida ainult kooli infojuht. Pärast seda kui rakenduse administraator on määranud kasutaja kooli infojuhi rolli, saab kasutaja kooli registreerida. Kooli registreerimisel määrab infojuht perioodi või perioodid, mida kasutatakse pikaajalisite broneeringute tegemiseks. Infojuht määrab ka kooli tunniajad, millal õpetajad broneeringuid saavad teha ja samuti klassid, mida rakenduses broneerida on võimalik.
4. Kooli andmete muutmine – pärast kooli registreerimist peab olema võimalik muuta perioodide, tunniaegade ja klasside andmeid.

5. Kasutajate haldamine – kooli kasutajaid saab hallata ainult kooli infojuht. Infojuht saab lisada, muuta ja kustuda kasutajaid, kes on valinud infojuhi poolt registreeritud kooli kasutajaks registreerimisel. Infojuht saab anda kasutajatele õigused klassi broneerimiseks rakenduses.
6. Klassi broneerimine – õpetaja kasutajagruppi määratud kasutajad saavad broneerida klasse. Broneerida saab ainult klasse, mis on määratud infojuhi poolt. Samuti saab klasse broneerida ainult infojuhi poolt määratud tunniaegadeks. Broneeringu tegemisel on võimalik valida õppeaine ja lisada märkuseid broneeringu kohta.
7. Pikaajaline broneering – õpetajatel peab olema võimalus broneerida ka klassi pikemaks perioodiks. Eelkõige mõeldud informaatikaõpetajatele, kelle tunnid peamiselt arvutiklassides toimuvad. Pikaajalise broneeringu tegemise periood määratakse infojuhi poolt.
8. Broneeringu konfliktid – õpetajad peavad saama broneerida klassi samaks ajaks. Ühe ja sama klassi samaaegse broneeringu korral tekib konflikt, mis lahendatakse läbirääkimiste teel e-maili kaudu ja mille tulemusena jääb ainult üks broneering.
9. Nädalavaate printimine – õpetajatel ja kooli infojuhil on võimalus printida nädala kaupa broneeringuid PDF formaadis.
10. Statistika kuvamine – kooli direktorile kuvatakse statistikat klasside broneerimise kohta.
11. Broneeringute eksport – tehtud broneeringuid on võimalik eksportida CSV faili statistika eesmärgil.
12. Sisselogimine Facebook'i abil.

3.3 Andmemudel

Rakenduse andmebaasis on autori poolt loodud järgmised tabelid:

Joomla kasutajad – j8sdv_users tabel andmebaasis. Sisaldab andmeid kasutajatest ja mida kasutatakse rakenduses autentimiseks.

Kasutajagrupp – j8sdv_usergroups tabel andmebaasis. Sisaldab andmeid kasutajagruppide kohta.

Kasutaja grupis – j8sdv_usergoup_map tabel andmebaasis. Määrab kasutajatele kasutajagrupid.

Kasutajad – tabelis on andmed rakenduse kasutajate kohta.

Koolid – tabelis on koolide nimekiri ja neid koole saab registreerida rakenduses. Nimekiri on saadud Eesti Hariduse Infosüsteemist(EHIS).

Registreeritud – tabelis on andmed registreeritud koolide kohta.

Periood – tabelis on andmed perioodide kohta. On seotud registreeritud tabeliga, kus parent_id väljaga antakse registreerimised id. Selle abil saab väljaselgitada, millised kooli perioodide on tegemist.

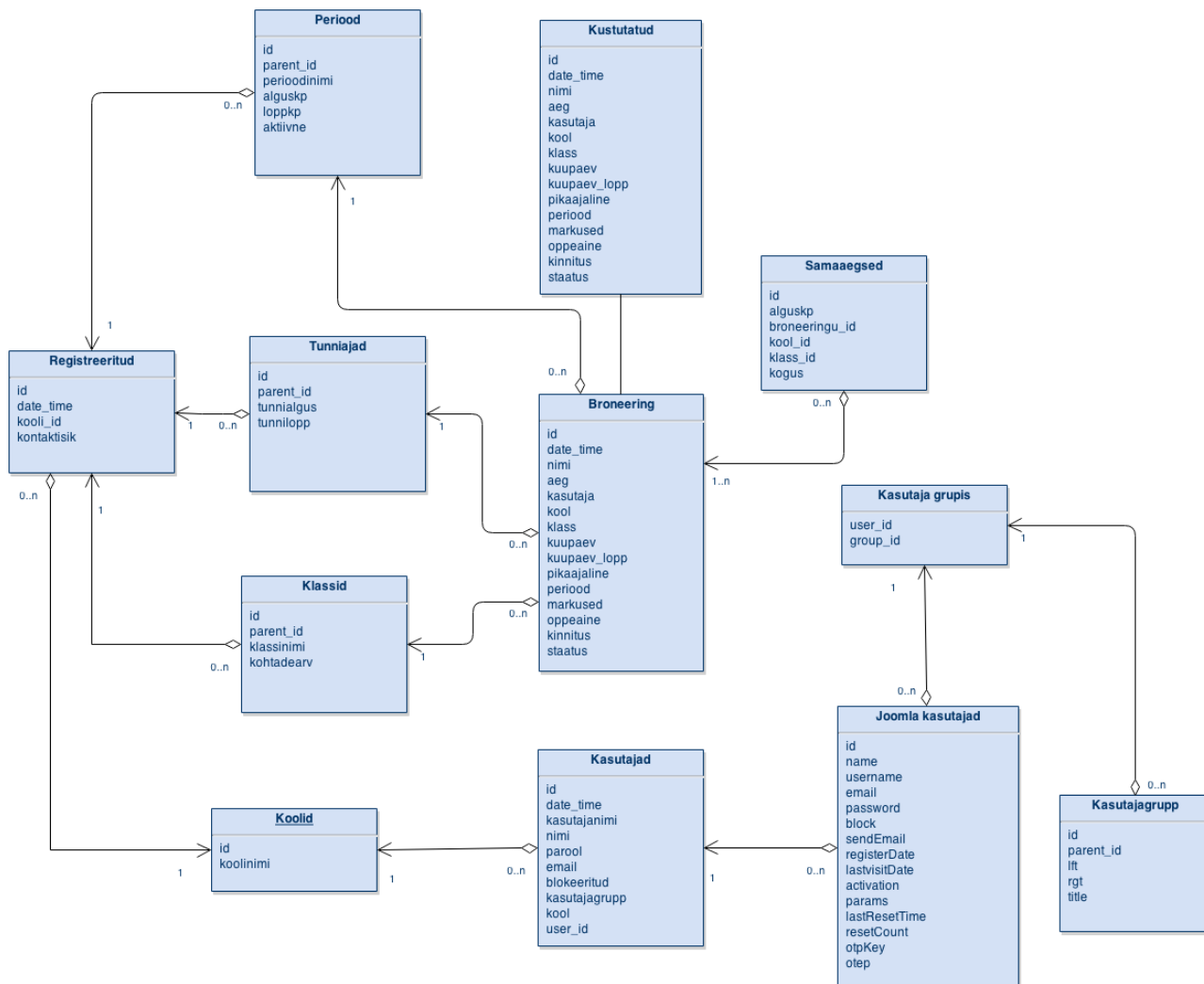
Tunniajad – tabelis on andmed tunniaegade kohta. Sarnaselt perioodiga on tabel seotud registreeritud tabeliga.

Klassid – tabelis on andmed klasside kohta. Sarnaselt tunniaegade ja perioodide tabeliga on klasside tabel seotud registreeritud tabeliga.

Broneering – tabel sisaldab andmeid broneeringute kohta.

Kustutatud – tabel sisaldab andmeid konfliktide korral kustutatud broneeringutest. Tabelis on samad väljad mis broneeringu tabelis.

Samaaegsed – tabelis on andmed samaaegsete broneeringute kohta. Vajalik üheks ja samaks ajaks tehtud broneeringute koguse väljaselgitamiseks.



Joonis 2 – Rakenduse andmemudel

3.4 Realiseerimine

Nagu eelpool mainitud kasutatakse rakenduse loomisel Fabrik komponenti. Rakenduse tegemine sai alguse Fabrik komponendis list’ide loomisest. List’e võib vaadata kui andmebaasi tabelite vaateid, millele saab erinevaid parameetreid määrata (näiteks järjestus, andmete filtreerimine, ligipääsu õigused jne). (Vt Joonis 3)

ID	List name	Db table	Element	Form	View data	View details	Published
11	Kasutajad	kasutajad	+ Add	✎ Edit	☰ View data	View details	✓
14	Klassi broneerimine	broneeringud	+ Add	✎ Edit	☰ View data	View details	✓
4	Kooli registreerimine	registreeritud	+ Add	✎ Edit	☰ View data	View details	✓
15	Kooli sätted	registreeritud	+ Add	✎ Edit	☰ View data	View details	✓
1	Koolid	koolid	+ Add	✎ Edit	☰ View data	View details	✓
8	Registreerimine	kasutajad	+ Add	✎ Edit	☰ View data	View details	✓

Joonis 3 – Fabrik. Listide nimekiri

List'ide loomisel luuakse ka vastavad vormid, mille abil kasutajatelt andmeid saadakse. Vormile saab lisada elemente. Elemendid on vormi väljad, mis andmetabelis veerge tähistavad.

3.4.1 Kasutajate registreerimine ja haldamine

Kasutajate registreerimiseks ja haldamiseks luuakse kõigepealt kasutajate list, mis kasutajate andmeid hoidma hakkab. Registreerimise vormile on lisatud kasutaja kohta käivad elemendid.

ID	Name	Label	Full Element Name	Validations	Group	Plugin	Show in List	Published
54	id	id	kasutajad__id	0/0	Registreerimine	internalid	✓	✓
55	date_time	date time	kasutajad__date_time	0/0	Registreerimine	date	✓	✓
56	username	Kasutajanimi	kasutajad__username	0/0	Registreerimine	field	✗	✓
57	name	Nimi	kasutajad__name	0/0	Registreerimine	field	✗	✓
58	password	Parool	kasutajad__password	0/0	Registreerimine	password	✗	✓
59	email	E-mail	kasutajad__email	0/0	Registreerimine	field	✗	✓
60	block	Blokeeritud	kasutajad__block	0/0	Registreerimine	yesno	✗	✓
61	user_group	Kasutajagrupp	kasutajad__user_group	0/0	Registreerimine	databasejoin	✗	✓
62	school	Valige kool	kasutajad__school	0/0	Registreerimine	databasejoin	✗	✓
72	user_id	user id	kasutajad__user_id	0/0	Registreerimine	field	✗	✓

Joonis 4 – Fabrik. Registreerimise vormi elemendid

Autentimine Joomla! toimib kasutades Joomla! kasutajate andmetabelit j8sdv_users. Fabrik pakub võimalust seostada Fabrik komponendi abil loodud vormi seostada Joomla! kasutajate andmetabeliga. Lisades registreerimise vormile „juser“ plugina, mille abil väljad Joomla! kasutajate andmetabeliga seostatakse. (Vt Joonis 5)

Binds data to a Joomla user account

Fields [Bypass Joomla settings](#) [Options](#) [Sync](#)

Name field	<input type="text" value="Nimi"/>
Username field	<input type="text" value="Kasutajanimi"/>
Password field	<input type="text" value="Parool"/>
Email field	<input type="text" value="E-mail"/>
Block field	<input type="text" value="Blokeeritud"/>
User ID field	<input type="text" value="user id"/>
User group field	<input type="text" value="Kasutajagrupp"/>

Joonis 5 – Fabrik. Andmete sidumine Joomla kasutajatega kasutades „juser“ pluginat

Uue kasutaja registreerimisel ja olemasoleva kasutaja muutmisel läbi Fabrik poolt loodud registreerimise vormi lisatakse ja muudetakse vastavaid andmeid Joomla kasutajate andmetabelis.

Kasutajate registreerimine ja kasutajate haldamine kasutavad küll sama andmetabelit, kuid kasutajate haldamist peaks saama kasutada rakenduses ainult kooli infojuhi rolli määratud kasutajad. Selleks tuleb luua kasutajate haldamise kohta eraldi list. Andmed salvestatakse küll samase andmetabelisse, kuid kasutajate haldamisele määratakse teistsugused ligipääsuõigused. Kooli infojuht peaks saama hallata ainult sama kooli kasutajaid. Selleks määratakse kasutajate list'ile filter, mis esitab ainult kasutajad, kelle on sama kool, mis sisselogitud kasutajalgi. (Vt Joonis 6)

Prefilters

Join	Field	Condition	Value	Type	Apply to	Grouped	Delete
WHERE	Valige kc	EQUALS	<pre>\$user = JFactory::getUser(); \$currentId = \$user->id; \$db = JFactory::getDbo(); \$query = \$db->getQuery(true); \$query="select * from kasutajad where user_id =\$currentId; \$db->setQuery(\$query); \$row = \$db->loadObject(); return \$row->school;</pre>	Eval	Administ	n/a	<input type="button" value="X"/>

Joonis 6 – Fabrik. Kasutajate list'ile filtri lisamine

Kasutajate haldamise vormile on samamoodi lisatud „juser“ plugin ja väljad seotud Joomla kasutajate andmetabeli väljadega. Autoril ei õnnestunud siduda kasutajagrupperi välja Joomla kasutajagrupperidega. Kasutajate haldamisel kasutajagrupperi muutmisel kuvatakse veeteade. Autor lahendas probleemi eemaldades „juser“ pluginas kasutajagrupperi seose ning uuendab käsitsi peale vormi sisestamist Joomla kasutajagrupperi tabelit vormist saadud väärtusega.

```
$userid = $formModel->getElementData('kasutajad__user_id', true);
$gid = $formModel->getElementData('kasutajad__user_group', true);
foreach ($gid as $val){
    if (!empty($userid) && !empty($val)) {
        $db = JFactory::getDbo();
        $query = $db->getQuery(true);
        $query="UPDATE j8sdv_user_usergroup_map SET group_id=".$val." WHERE user_id=".$userid;
        $db->setQuery($query);
        $found = (int) $db->execute();
    }
}
```

Koodinäide 1 – Fabrik. Kasutajagrupperi muutmise

3.4.2 Broneeringu tegemine

Broneerimise funktsionaalsuse lisamiseks loodi broneeringute list ja määrati broneeringu väljad (lisati elemendid). (Vt Joonis 7)

ID	Name	Label	Full Element Name	Validations	Group	Plugin	Show in List	Published
114	id	id	broneeringud__id	0/0	Klassi broneerimine	internalid	<input type="checkbox"/>	<input checked="" type="checkbox"/>
115	date_time	date time	broneeringud__date_time	0/0	Klassi broneerimine	date	<input type="checkbox"/>	<input checked="" type="checkbox"/>
128	oppeaine	õppeaine	broneeringud__oppeaine	0/0	Klassi broneerimine	dropdown	<input type="checkbox"/>	<input checked="" type="checkbox"/>
116	nimi	Broneeringu nimi	broneeringud__nimi	0/0	Klassi broneerimine	field	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
117	aeg	Aeg	broneeringud__aeg	0/0	Klassi broneerimine	dropdown	<input type="checkbox"/>	<input checked="" type="checkbox"/>
121	kuupaev	Kuupäev	broneeringud__kuupaev	0/0	Klassi broneerimine	date	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
118	kasutaja	kasutaja	broneeringud__kasutaja	0/0	Klassi broneerimine	field	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
119	kool	kool	broneeringud__kool	0/0	Klassi broneerimine	field	<input type="checkbox"/>	<input checked="" type="checkbox"/>
120	klass	Klass	broneeringud__klass	0/0	Klassi broneerimine	dropdown	<input type="checkbox"/>	<input checked="" type="checkbox"/>
122	kuupaev_lopp	Lõpp	broneeringud__kuupaev_lopp	0/0	Klassi broneerimine	date	<input type="checkbox"/>	<input checked="" type="checkbox"/>
124	pikaajaline	Pikaajaline broneering	broneeringud__pikaajaline	0/3	Klassi broneerimine	yesno	<input type="checkbox"/>	<input checked="" type="checkbox"/>
125	kasutajagrupp	kasutajagrupp	broneeringud__kasutajagrupp	0/1	Klassi broneerimine	field	<input type="checkbox"/>	<input checked="" type="checkbox"/>
126	periood	periood	broneeringud__periood	0/0	Klassi broneerimine	dropdown	<input type="checkbox"/>	<input checked="" type="checkbox"/>
127	markused	Märkused	broneeringud__markused	0/0	Klassi broneerimine	textarea	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Joonis 7 – Fabrik. Klassi broneerimise elemendid

Peale iga broneeringu tegemist kui broneeringu andmed on juba andmebaasi salvestatud, muudetakse kuupäeva ja genereeritakse kinnituse kood. Kalendris kuvatakse broneering alguskuupäeva ja lõppkuupäeva elementide järgi. Selleks tuleb andmetabelis kuupäevale lisada ka aeg. Selleks võetakse kasutaja poolt määratud ajavahemik ja lisatakse tunni alguse kellaeg alguskuupäevale (elemendi nimi: „kuupaev“) ja tunni lõpu kellaeg lõppkuupäevale (elemendi nimi: „kuupaev_lopp“). Seejärel uuendatakse andmetabelis vastavate veergude andmeid.

```
$ajad = explode(" - ", $ajavahemik); // $ajavahemik – broneerimise vormis määratud aeg. Nt: 09:00:00 – 10:45:00
```

```
$esimene = $ajad[0]; // tunni algus
```

```
$steine = $ajad[1]; // tunni lõpp
```

```
$algusajad = explode(":", $esimene);
```

```
$loppajad = explode(":", $steine);
```

```
$algus1 = $algusajad[0]; // tunni alguse tund. Nt: 09
```

```
$algus2 = $algusajad[1]; // tunni alguse minut. Nt: 00
```

```
$lopp1 = $loppajad[0]; // tunni lõpu tund. Nt: 10
```

```
$lopp2 = $loppajad[1]; // tunni lõpu minut. Nt: 45
```

```
$kuupaev = explode("-", $kp); // $kp – broneerimise vormis määratud kuupäev. Nt: 2014-05-03
```

```
$alguskp = date("Y-m-d H:i:s", mktime($algus1, $algus2, 0, $kuupaev[1], $kuupaev[0], $kuupaev[2]));
```

```
$loppkp = date("Y-m-d H:i:s", mktime($lopp1, $lopp2, 0, $kuupaev[1], $kuupaev[0], $kuupaev[2]));
```

```
// Kinnituskoodi genereerimine, mida kasutatakse broneeringu konflikti korral kustutamisel.
```

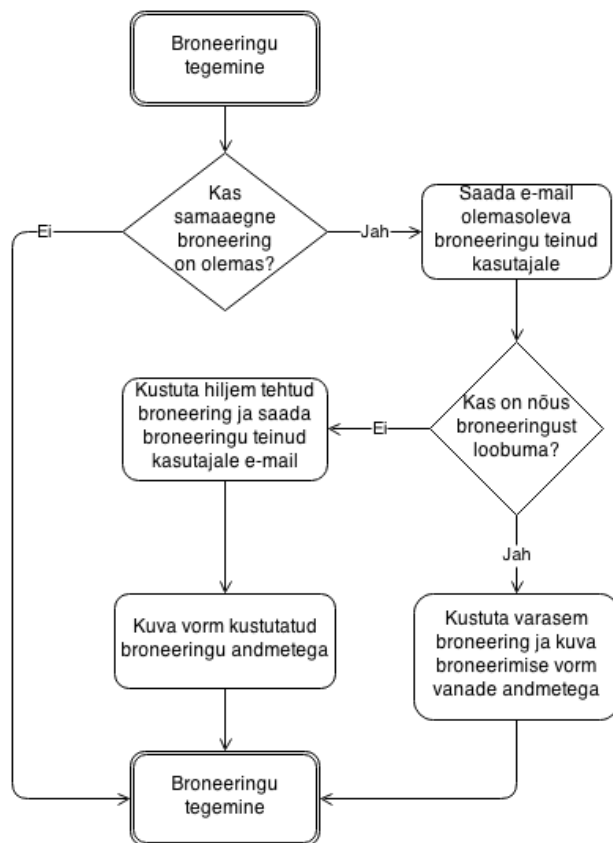
```
$confirmation = md5(uniqid(rand(), true));
```

Koodinäide 2 – Fabrik. Tunniaja lisamine kuupäevale

Rakenduses saab teha kahte tüüpi broneeringut: ühekordne ja pikaajaline. Pikaajalise broneeringu valimisel kuvatakse perioodi väli (muidu on peidetud), milles on aktiivsete perioodide nimekiri. Valitud perioodi vahemikus on võimalik pikaajalist broneeringut sooritada. Perioodid on määranud kooli infojuht. Kui broneerimisel valitud kuupäev ei jää pikaajalise perioodi vahemikku, siis broneeringut ei tehta ja esitatakse veateade. See on realiseeritud lisades vastav kuupäevade kontroll PHP skripti abil enne andmebaasi salvestamist.

3.4.3 Konflikti lahendamine

Disainisessiooni käigus selgus, et samaaegsete broneeringute konflikt tuleks rakenduses lahendatud õpetajate vaheliste läbirääkimiste teel. Iga õpetaja peaks saama sama klassi juba olemasolevale ajale broneerida, kuid õpetaja kes broneeris klassi varem otsustab, kas loovutab oma broneeritud aja või keeldub. Keeldumise korral tühistatakse hiljem tehtud broneering. (Vt Joonis 8)



Joonis 8 – Konflikti lahendamise skeem

Broneeringu konflikt on lahendatud Fabrik'us lisades klassi broneerimise vormile PHP skripti, mis käivitub peale broneerimise vormi sisestamist kui andmed on juba andmebaasi sisestatud (ingl k. *End of form submission*). Kood kontrollib, kas samaaegseid broneeringuid juba eksisteerib.

```

$query = $db->getQuery(true);

$query = 'SELECT * FROM broneeringud WHERE kool='.$kool.' AND aeg="'.$aeg.'" AND
klass="'.$klass.'" AND kuupaev="'.$alguskp.'" AND id NOT IN ( '.$id.' )';

$db->setQuery($query);

$total = $db->loadResult();

if ($total > 0) {

    // KONFLIKTI E-MAILI SAATMINE

}

```

Koodinäide 3 – Fabrik. Konflikti korral e-maili saatmine

Kui broneerimisel tekkis konflikt, siis saadetakse varem broneerimise teinud õpetaja e-mail kahe lingiga, millega otsustab, kas loovutab oma broneeringu või mitte.

Õpetaja Peeter (peeter@broneering.bl.ee) soovib broneerida klassi: A-317 kuupäeval: 29.05.2014 tunniajaks: 9:00 – 9:45. Kas olete nõus oma broneeringu aja loovutama?

Nõustu:

<http://broneering.bl.ee/index.php/klassi-broneerimine/form/14?action=accept&id=389&confirmation=3ebf3236bfba8773f45eda1370669eb0>

Keeldu:

<http://broneering.bl.ee/index.php/klassi-broneerimine/form/14?action=decline&id=392&confirmation=271846c07017ea0a5cb5d366769d70a1>

Koodinäide 4 – Konflikti korral saadetud e-maili näidis

Linkidega antakse kaasa kinnituskood, mis genereeritakse iga broneeringu tegemisel. Broneeringu kustutamisel kontrollitakse, kas URL-ist saadud kinnituskood ja andmebaasis olev kinnituskood vastava broneeringu real on sama. Sellega välistatakse, et kasutajad ainult broneeringu „id“ järgi URL-i kaudu broneeringuid kustutada ei saaks.

Nõustu lingile vajutades kustutatakse e-maili saanud õpetaja broneering ning talle esitatakse broneeringu vorm, kus väljad on täidetud kustutatud broneeringu andmetega. Kasutaja peaks muutma kas klassi või muutma broneeringu aega (kuupäeva või tunniaega).

Keeldu lingile vajutades kustutatakse hiljem tehtud broneering ning broneeringu teinud kasutajale saadetakse e-mail teavitamiseks broneeringu tühistamisest. Talle saadetakse ka link, mille kaudu saab uue broneeringu teha, kus väljad on täidetud tühistatud broneeringu andmetega. Kasutaja peaks siis järjekordselt muutma kas klassi või broneeringu aega (kuupäeva või tunniaega).

4 Evalvatsioon

Käesoleva töö raames loodud rakendust hinnates leiab autor, et suurem osa rakenduse nõudmisi sai edukalt täidetud. Rakenduse kasutamine on mugav ja lihtne ning rakenduse abil on võimalik koguda andmeid tehtavate broneeringute kohta. Kuigi andmete eksportimise funktsionaalsus jäi kasutajaliideses veel realiseerimata, on seda võimalik teha rakenduse administratsiooniliideses (*back-end*'is). Järelikult sai täidetud ka rakenduse põhieesmärk.

Rakendus on küll prototüübi faasis, kuid suurem ja olulisem osa funktsionaalsusi sai realiseeritud. Realiseerimata jäid järgnevad funktsionaalsused:

- Nädalavaate printimine.
- Kooli direktorile statistika kuvamine.
- Broneeringu andmete eksport.
- Facebooki sisselogimise moodul.

Hilisemal hindamisel ja testimisel on väljatulnud ka mõned puudused.

Ülevaade puudustest:

- Kooli registreerimine on ühekordne tegevus, seega pärast kooli registreerimist ei tohiks enam saada registreerida. Hetkel on veel see viga parandamata.
- Klassi broneerimise kalendri kuuvaates esineb viga, kui broneeringu tegemiseks teha topeltklõps kalendris, siis kuupäeva väärtuseks tuleb eelmise päeva kuupäev. Samuti kalendri nädala- ja päevavaates, kui teha broneeringu lisamiseks topeltklõps ajavahemikus 00:00, kuni 03:00, siis esitatavas vormis kuvatakse kuupäeva väljal eelmise päeva kuupäev.
- Õpetaja saab sooritada topeltbroneeringut ehk tekitada konflikti iseenda broneeritud ajaga.

5 Edasiarendamise võimalused

Enne rakenduse väljastamist tuleks kõigepealt puuduolevad funktsionaalsused lisada ja parandada teadaolevad vead. Rakendust tuleks ka põhjalikumalt testida ja kui vaja, siis teha uus disainisessioon ja katsetada uuesti ekspertide peal rakendust. Rakenduse kood tuleks ülevaadata ja vajadusel optimiseerida.

Rakendus kasutab Joomla vaikimisi disainimalli. Rakendusele tuleks lisada ka kujundus või kohandada olemasolevat kujundust. Seda saab lihtsasti teha, kasutades Joomla disainimalli laiendust. Käesolevas töös loodud rakenduse eesmärgiks ei olnud rakendust kujundada.

Rakendus on mõeldud arvutiklasside broneerimiseks, kuid rakenduse kasutust võib laiendada ka teistele klassidele. Sellisel juhul tuleks rakendusele lisada klassile tüüp, sest rakenduse eesmärgiks on koguda andmeid arvutiklasside kasutamise kohta ja samuti tuleks optimiseerida kalendrit suure arvu klasside broneeringute esitamiseks.

Rakendusele võiks ka lisafunktsionaalsusi kasutajate tagaside põhjal luua. Rakenduse põhieesmärgiks on lihtsus, seega funktsionaalsustega üritati mitte üle pingutada.

Kokkuvõte

Bakalaureuse töö tulemusena anti ülevaade olemasolevatest broneerimisrakendustest ning nende eelistest ja puudustest töös loodava rakenduse vajadustest lähtudes, koostati rakenduse disain ja loodi prototüüp disaini alusel. Bakalaureusetöö eesmärk, luua rakendus, mis võimaldab Eesti koolide õpetajatel arvutiklasse broneerida, on saavutatud. Rakenduse abil on võimalik koguda andmeid Eesti koolide arvutiklasside kasutamise kohta. Rakendusest valmis prototüüp. Esialgsel funktsionaalsuste testimisel tulid välja ka mõned vead, millest kõiki ei jõutud parandada töö esitamise tähtajaks.

Rakenduse disainimine toimus bakalaureuse töö raames. Disainimise tulemusena saadud olulisemad funktsionaalsed nõuded täideti. Puuduolevate funktsionaalsuste lisamine ja olemasolevate vigade parandamine on plaanitud rakendada pärast bakalaureusetöö valmimist.

Loodud veebirakendus töötab Windowsi ja Linuxi operatsioonisüsteemide peal ning tehnilised nõuded vastavad Joomla tehniliste nõuetega. [6] Töö esitamise hetkel on rakendus ka avalikult üleväl:

- Broneerimisrakenduse kasutajaliides asub aadressil <http://broneering.bl.ee>;
- Rakenduse administraatoriliides asub aadressil <http://broneering.bl.ee/administrator>; kuhu ligipääs on võimalik koos töö autoriga

Summary: Computer classroom scheduling web application for Estonian schools

The aim of this B.A thesis was to develop a web application that allows Estonian teachers to book computer classrooms. The premise is that currently booking in schools is done using old-fashioned methods. Development process started by designing the web application. This was done using scenario-based method. The outcome of the design process was a list of functionalities that the web application should implement. Based on those functionalities a prototype was made. Also one of the web application's goals is to collect bookingdata for research purposes.

The thesis consists of 5 chapters. The first chapter provides an overview of web applications that have already been made for booking and scheduling purposes. The second chapter describes the design methods and gives an overview of the design process. The third chapter describes the web application and how some of the functionalities were implemented. The forth chapter gives an evaluation of the results and fifth chapters gives suggestions for future development.

Web application is still in prototype phase and some of the functionalites have not been implemented yet. Also there are some issues that have not been fixed. Despite that, the main functionalities work and the purpose is to further develop the web application and realease it in the future.

Kasutatud kirjandus

1. Asio-Data Oy(2013). Asio EduERP Classroom Booking System. Asio EduERP whitepaper, 4-7. Kasutamise kuupäev: 2. mai 2014. a. Allikas Asio EduERP lühiülevaade: http://www.Asio.eu/Asio/english/Asio_ohjelmat_white_paper_eng.pdf
2. Asio EduERP Classroom Booking. Kasutamise kuupäev: 2. mai 2014. a. Allikas Asio koduleht: http://www.Asio.eu/Asio/english/classroom_booking.shtml
3. Configuration. Classroombookings documentation. Kasutamise kuupäev: 2. mai 2014. a. Allikas Classroombookings rakenduse koduleht: <http://classroombookings.com/documentation/v1.0/configuration>
4. Fabrik Wiki. Kasutamise kuupäev: 2. mai 2014. a. Allikas Fabrik koduleht: <http://fabrikar.com/forums/index.php?wiki/index/>
5. Joomla documentation. Extension types (general definitions). Kasutamise kuupäev : 2. mai 2014. a. Allikas Joomla dokumentatsiooni koduleht: [http://docs.joomla.org/Extension_types_\(general_definitions\)](http://docs.joomla.org/Extension_types_(general_definitions))
6. Joomla Technical Requirements. Kasutamise kuupäev: 2. mai 2014. a. Allikas Joomla dokumentatsiooni koduleht: http://docs.joomla.org/Technical_requirements
7. Michelle Lindblom (26. september 2012. a.), Doodle: Easy Online Scheduling. Kasutamise kuupäev: 4. mai 2014. a. Allikas: wiredimpact.com: <http://wiredimpact.com/blog/doodle-easy-online-scheduling/>
8. Multi-View Calendar for Joomla. Kasutamise kuupäev: 2. mai 2014. a. Allikas Multi-View Calendar komponendi koduleht: <http://www.joomlascalendars.com/>
9. Rosson, M. B, & Carrol, J. M. (2002). Scenario-Based Design. Chapter 53 in J. Jacko & A. Sears (Eds.), The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications. Lawrence Erlbaum Associates, 2002, lk. 1032-1050. Kasutamise kuupäev: 4. mai 2014. a. URL: https://www.e-education.psu.edu/drupal6/sites/www.e-education.psu.edu/geog583/files/Rosson_2002_sbd.pdf

10. What is Joomla? Kasutamise kuupäev: 2. mai 2014. a. Allikas Joomla koduleht:
<http://www.joomla.org/about-joomla.html>

LISA

Lisa 1. Persoon 1: Peeter



Vanus: 30

Haridus: Haridusteaduse magister

Amet: Infojuht

Peeter

Peeter on kooli infojuht. Ta on infojuhina töödanud 3 aastat. Tal on tugev IT-alane taust ja haridus.

Infojuhina töödab Peeter üritanud kooli kaasaegsemaks muuta. Kuna Peetri üheks ülesandeks on ka arvutiklassidega tegelemine ja üheks mureks oli nende kasutamisega seotud probleemid, siis tegi Peeter ettepaneku kasutada koolis arvutiklasside broneerimise rakendust.

Eesmärgid:

Töömahu vähendamine: Peeter ei sooviks tegeleda nii palju arvutiklasside broneerimisega kui sellega saab hakkama rakendus.

Kaasaegsus: Peeter leiab, et koolis kasutatav meetod broneerimiseks on aegunud ja IT taustaga inimesena leiab, et kogu tegevust saaks mugavamaks ja kaasaegsemaks muuta.

Foto 1. Autor: Yuri Samoilov. [https://www.flickr.com/photos/yuri_samoilov/9975661323/]

Lisa 2. Persoon 2: Eve



Vanus: 28

Haridus: Informaatikaõpetaja magister

Amet: Informaatikaõpetaja

Eve

Eve on noor informaatikaõpetaja, kes on õpetajana töödanud kõigest aasta.

Eva töötab enamuse ajast arvutiklassis. Tihti soovivad teised õpetajad samuti kasutada arvutiklassi. Selleks, et teiste õpetajate klassi kasutamine Eve tööd segama ei hakkaks ja üldse arvutiklassi paremaks kasutamiseks sooviks Eve kasutada broneerimise rakendust.

Eesmärgid:

Mugavam kasutamine: Kuna arvutiklassis on Eve põhiliseks kasutatavaks ruumiks, siis olemasoleva süsteemiga peab Eve iga nädala tunnid eraldi kõikidele nädalatele ära broneerima. Rakenduse kasutamise puhul muutuks see lihtsamaks.

Muudatustele reageerimine: Kuna tunniplaan võib muutuda, siis on ka Evel lihtsam rakenduse kasutamisel broneeringu aegasid muuta.

Foto 2. Autor: Frank Tasche. [https://www.flickr.com/photos/f_tasche/8869999569/]

Lisa 3. Persoon 3: Linda



Vanus: 45

Haridus: Loodusteaduste
bakalaureus

Amet: Geograafiaõpetaja

Linda

Linda on 25 aastase kogemusega geograafiaõpetaja. Ta õpetab mitmes Eesti koolis.

Et kaasaegsemat haridust õpilastele anda soovib Linda aegajalt arvutiklasse kasutada. Ühes koolis, kus Linda õpetab on kasutusele võetud arvutiklasside broneerimise rakendus, mida Linda on soovitanud ka teistele koolidele, kus tema õpetab.

Eesmärgid:

Lihtsus: Linda jaoks on väga lihtne broneerida arvutiklassi.

Foto 2. Autor: JAXPORT. [<https://www.flickr.com/photos/jaxport/9569544177>]

Lisa 4. Persoon 4: Juhan



Vanus: 58

Haridus: Magistrikraad

Amet: Direktor

Juhan

Juhanil on strateegilise juhtimise magistrikraad. Juhan on koolidirektor olnud 14 aastat. Juhan sooviks saada erinevat statistikat koolis olevate arvutiklasside kasutamise kohta. Infojuhi soovitusel otsustas Juhan koolis kasutusele võtta arvutiklasside broneerimise rakenduse.

Eesmärgid:

Efektiivus: Juhan sooviks, et kooli õpetajatel oleks võimalik kasutada arvutiklasse õppetöös ning et see ei segaks informaatikaõpetaja tööd.

Statistika: Soov on saada parem ülevaade arvutiklasside kasutamise kohta.

Foto 4. Autor: Eliot Kimber. [<https://www.flickr.com/photos/woods-kimber/1440499238>]

Lisa 5. Stsenaarium 1: Kooli registreerimine

Peeter on noor kooli infojuht ühes Eesti suures koolis. Tihti pöörduvad kooli õpetajad Peetri poole sooviga broneerida arvutiklassi tundide jaoks. Peeter kasutab selleks Google Docs'i Spreadsheet lahendust.

Peeter sõidab rongiga tööle ja loeb e-maili. Üks tema koolis töötav õpetaja on Peetrile kirjutanud e-maili, soovitamaks Peetril kasutada klasside broneerimiseks rakendust. Peeter tunneb asjast huvi ja registreerib ennast rakenduse kasutajaks ja saadab rakenduse administraatorile e-maili sooviga registreerida kool, kus Peeter töötab, rakendust kasutama. Pärast administraatori kinnitust registreerib Peeter kooli. Kooli registreerimisel määrab Peeter ära millist perioodisüsteemi kooli kasutab ja määrab perioodide ajad. Samuti määrab Peeter tundide ajad koolis ning määrab ka klassid, mida rakenduse abil broneerida on võimalik. Peeter kirjutab e-maili saatnud õpetajale tagasi, teavitades et Peeter on registreerinud kooli rakendust kasutama.

Pärast kooli registreerimist on Peetril võimalus lisatud andmeid muuta ja ka ise broneeringut teha, juhuks kui kooli arvutiklassi soovib kasutada külalisõpetaja. Peeter näeb ka registreeritud kasutajaid, kes on valinud sama kooli, kus Peeter töötab. Peeter saab määrata kasutajatele kas „õpetaja“ või „informaatikaõpetaja“ rolle, millega antakse broneerimisel eelis.

Lisa 6. Stsenaarium 2: Ühekordne broneering

Linda on kogenenud geograafia õpetaja, kes õpetab kahes erinevas koolis. Aegajalt on Lindal vaja kasutatada arvutiklasse teatud õppematerjali õpilastele paremini seletamiseks. Ühes koolis, kus Linda õpetab kasutatakse arvutiklasside broneerimiseks broneerimise rakendust. Linda soovitas rakendust ka teisele koolile ning nüüd kasutatakse rakendust ka sealgi. Linda sai infojuhilt e-maili rakenduse kasutamise kohta ja registreeris ennast rakenduses selle kooli nimekirja.

Rakenduses kooli registreerinud infojuht lisas Linda õpetajate hulka, kes arvutiklasse broneerida saavad.

Kuna Linda kasutab arvutiklasse harva, siis saab tema rakenduses teha ainult ühekordset broneeringut. Rakenduses saab Linda valida millise kooli arvutiklassi ta soovib broneerida. Linda valib klassi ja ajavahemiku (tunniaja), millal broneeringut soovib teha.

Lisa 7. Stsenaarium 3: Pika-ajaline broneering

Eve on noor informaatikaõpetaja ühes Eesti gümnaasiumism, mis kasutab arvutiklassi broneerimiseks rakendust. Eve saab iga veerandi alguses õppelajuhtaja poolt koostatud tunniplaani ja samuti on õppealajuhataja palunud Evel broneerida arvutiklass tunniplaani kohaselt. Eve registreerib ennast rakendust kasutama täites vajalikud väljad ning valides nimekirjast kooli. Pärast seda kui kooli infojuht on Eve kasutaja aktiveerinud saab Eve rakenduses klasse broneerima hakata.

Kuna Eve kasutab informaatika õpetamiseks ainult arvutiklasse, siis Eve saab broneerida arvutiklassi terveks veerandiks. Koolis, kus Eve õpetab on kooli infojuht lisanud rakendusse Eve õpetajaks, kellel on võimalik pikaajalisi broneeringuid teha. Eve valib päeva, klassi ja ajavahemiku (tunniaja) broneeringu tegemiseks ning rakendus pakub Evel valikut broneerida klass terveks perioodiks, mis määratud kooli infojuhi poolt. Pärast broneeringu tegemist näeb Eve kalendris oma broneeringut ja broneering on tehtud terve perioodi vältel.

Lisa 8. Stsenaarium 4: Konflikti lahendamine

Informaatikaõpetaja Eve ja geograafia õpetaja Linda õpetavad samas koolis. Linda broneeris arvutiklassi ette ära, et saaks õpilastega geograafia olümpiaadiks harjutada. Kuna tunniplaanis toimus muudatus ja informaatika tunni aega muudeti, siis juhtus, et informaatika tund on samal ajal, millal Linda klassi broneeris. Eve broneerib arvutiklassi uue tunniplaani kohaselt ja Linda broneering tühistatakse, kuna kooli infojuhi poolt on määratud informaatikaõpetajale Evele prioriteet arvutiklassi broneerimisel. Lindale saadetakse rakenduse poolt E-mail, kus teavitatakse teda broneering tühistamisest.

Lisa 9. Stsenarium 5: Statistika

Juhan on kooli direktor. Kooli infojuhi soovitusel otsustas Juhan hakata koolis kasutama arvutiklasside broneerimiseks vastavat rakendust. Juhan registreeris ennast kasutajaks aga kuna Juhan on hajameelne ja unustab tihti paroole, siis logib Juhan rakendusse läbi Facebook'i. Juhan näeb rakenduses erinevat statistikat arvutiklasside ja nende broneerimise kohta. See võimaldab Juhanil teha muudatusi selliselt, et tulevikus arvutiklasse võimalikult efektiivselt kasutatakse.

Lisa 10. Lähtekood

Rakenduse lähtekood on saadaval kaasasoleva DVD-plaadi failis „lahtekood.zip“. Samuti on rakenduse kood üleval github-is: <https://github.com/raimov/broneering>

Rakenduse kasutamiseks tuleb importida andmebaasi fail „u508267162_kool.sql“, mis asub juurkataloogis. Samuti tuleb muuta rakenduse konfiguratsiooni faili „configuration.php“ (asub samamoodi juurkataloogis) vastavalt serveri andmetele.