

TARTU ÜLIKOOL  
MATEMAATIKA-INFORMAATIKATEADUSKOND

Arvutiteaduse Instituut

Infotehnoloogia eriala

Martin Suvorov

## Infosüsteem materjalide haldamiseks RAJU keskusele

Bakalaureusetöö (6 EAP)

Juhendajad: lektor Anne Villems

spetsialist Taavi Duvini

Autor: MARTIN SUVOROV.....“.....“ juuni 2011

Juhendajad: ANNE VILLEMS.....“.....“ juuni 2011

TAAVI DUVIN.....“.....“ juuni 2011

Lubada kaitsmisele

Professor: .....“.....“ juuni 2011

TARTU 2011

# Sisukord

Sissejuhatus .....	3
1. Robotite arendamise ja uurimise (RAJU) keskus ja selle seos robotika arendamisega Eestis .....	4
1.1. Robotika taust Eestis .....	4
1.2. Hetkel RAJU keskuse poolt kasutatavad infotehnoloogilised vahendid ja kogutud informatsioon .....	5
2. Uue infosüsteemi analüüs .....	8
2.1. Kasutajad .....	8
2.2. Nõuded .....	10
2.2.1. Kasutuslood .....	10
2.3. Struktuur .....	12
2.4. Riskid .....	16
3. Loodud süsteemi kirjeldus .....	17
3.1 Arhitektuur .....	17
3.2. Kasutajaliides .....	19
3.3. Suhtlus andmebaasiga .....	20
3.4. Süsteemi installeerimine .....	20
Kokkuvõte .....	23
Abstract .....	24
Viited .....	26
Lisad .....	28
Lisa 1. Süsteemi nõuded .....	28
Lisa 2. Kasutuslood .....	30
Lisa 3. Infosüsteemi lähtekood .....	39

## Sissejuhatus

2007. aastal alustas tegevust Kooliroboti projekt, mille eesmärgiks on edendada inseneriteadust Eesti koolides. Toetatakse robotikakursuste loomist, mille käigus õpilased saavad kokku panna ja programmeerida LEGO MINDSTORMS NXT roboteid. Õpetajatele korraldatakse koolitusi, koostatakse eestikeelseid õppematerjale ja töölehti.

Robotite uurimise ja arendamise (RAJU) keskus on Tartu Ülikooli arvutiteaduse instituudi juures tegutsev ühendus, mille eesmärgiks on LEGO MINDSTORMS NXT robotite ja nendega ühilduvate andurite uurimine ning selle põhjal õppematerjalide ja näidisülesannete koostamine. Paari aastaga on materjalide hulk jõudsalt kasvanud, kuid siiani puudub võimalus, kuidas neid kiiresti ja mugavalt õpetajatele ja õpilastele jagada.

Seetõttu otsustati sel aastal luua infosüsteem, mis haldaks kõiki olemasolevaid materjale. Selle infosüsteemi loomine ongi antud lõputöö eesmärgiks. Dokumendi võib jagada kolmeks suuremaks peatükiks - taustakirjeldus, infosüsteemi analüüs ja tulemuse kirjeldus.

Esimene peatükk annab üldise ülevaate robotika propageerimisega tegelevatest asutustest Eestis. Tutvustatakse RAJU keskust ja selle tähtsust. Samuti kirjeldatakse hetkel RAJU keskuse poolt kasutatavaid infotehnoloogilisi vahendeid.

Teine peatükk sarnaneb klassikalisele infotehnoloogilise projekti analüüsile - kirjeldatakse loodavas süsteemis olevaid kasutajaid ja nende võimalusi, süsteemi nõudeid ja süsteemi üldist struktuuri. Funktsionaalsed nõuded on kajastatud kasutuslugudena. Peatüki viimaseks punktiks on võimalike riskide hindamine.

Kolmas ja ühtlasi viimane peatükk annab ülevaate süsteemi ehitusest peale programmeerimise lõppu. Selles peatükis saab lugeja teada, mis programmeerimise keel on valitud süsteemi programmeerimiseks ja millist raamistikku kasutatakse. Kirjeldatakse süsteemi ehitust ning andmebaasi ja kasutajaga suhtlemist. Lõpuks on esitatud ka ülevaatlik kasutusjuhend süsteemi installeerimiseks.

# **1. Robotite arendamise ja uurimise (RAJU) keskus ja selle seos robotika arendamisega Eestis**

Enne kui on üldse võimalik rääkida RAJU keskusest ja selle tegemistest, tuleks selgitada, mis on robot ja robotika ning kes tegelevad sellega Eestis. Robot on programmeeritud masin, mida on võimalik kasutada inimese liikumist, tajumist ja mõtlemist asendavais töödes [1]. Robotite kasutamise põhjusi on mitmeid: nad suudavad tegutseda keskkondades, mis pole inimesele tööks sobivad, näiteks kõrge temperatuur, hapniku puudus või mõni muu oht elule ja tervisele. Samuti suudavad robotid teha efektiivsemalt nüri ja lihtsat tööd, mis kurnaks inimese kiiresti ära. Peale selle saab neid masinaid kasutada ka olukordades, kus inimese võimetest jääb puudu, näiteks on tarvis suuremat kiirust, tõstevõimet või täpsust. Neil põhjustel kasutatakse roboteid tänapäeval peaaegu igas valdkonnas ja tehnoloogia arenguga leitakse pidevalt uusi alasid, kus saab neid rakendada. Seetõttu on robotika ja masinaehitus väga tähtis ja perspektiivikas ala nii Eestis kui ka mujal maailmas. Robotika on teaduse haru, mis uurib robotite disaini, ehitamist ja kasutamist [2].

## **1.1. Robotika taust Eestis**

Robotika propageerimisega tegelevad hetkel Eestis neli suuremat organisatsiooni:

Tallinna Tehnika Ülikoolis (TTÜ) saavad tudengid õppida roboti- ja masinaehitust mehhatroonikainstituudis, kus uuritakse sügavamalt mehaanika-, elektroonika ja infotehnoloogiliste süsteemide koostoimet [3]. Lisaks tegutseb instituudi kõrval TTÜ Robotiklubi. TTÜ Robotiklubi on TTÜ tudengite poolt loodud klubi, mille eesmärgiks on arendada robotikaalaseid teadmisi ja oskusi tudengite seas [4]. Klubi liikmed osalevad enda loodud robotitega Robotex ja Baltic Robot Sumo võistlustel. Lisaks viiakse läbi kursust “Robot igapähele”, mille käigus antakse gümnaasiumiõpilastele võimalus ehitada ja programmeerida oma robot.

IT-Kolledži (ITK) Robotikaklubi on sarnaselt TTÜ Robotiklubiga ITK üliõpilaste organisatsioon, mille eesmärgiks on loengutes õpitu praktiline kasutamine loova töö käigus [5]. Ka nemad osalevad Robotex võistlustel.

Tartu Ülikooli (TÜ) Loodus- ja tehnoloogiateaduskonnas (LOTE) tegeldakse robotikaga ja mehhatroonikaga laiemalt professor Alvo Aabloo käe all. LOTE külje all tegutseb oma robotiklubi, kes pakub konkurentsi TTÜ Robotiklubile ja ITK Robotikaklubile Robotex võistlustel. 2007. aasta kevadel alustas tegevust Alvo Aabloo juhitud Kooliroboti projekt, mida toetab Tiigrihüppe Sihtasutus. Projekti püüab edendada inseneriteadust eesti koolides. Selleks kasutatakse LEGO MINDSTORMS NXT roboteid. Õpetajatele korraldatakse koolitusi ja õpilastele jagatakse eestikeelseid õppematerjale. Lisaks korraldatakse koolides robotiteatri etendusi. Koolide ülesandeks on leida sponsorid, kes aitaksid osta robotikomplekte. 2011. a. kevadeks oli projektiga liitunud juba 81 kooli [6].

RAJU keskus on mitteformaalne ühendus Tartu Ülikooli (TÜ) arvutiteaduse instituudis, mille koosseisu kuuluvad Anne Villems, Taavi Duvini ja TÜ tudengid. Anne Villemsi ja Taavi Duvini poolt juhitud tudengid koostavad LEGO MINDSTORMS robotite kohta erinevaid kursuseid, õppe- ja juhendmaterjale, õpetajatele mõeldud e-raamatuid ja ülesandeid erinevatele anduritele. Hiljem jagatakse need materjalid Kooliroboti projektiga liitunud koolide õpetajatele. Materjalide kaasabiga saavad õpetajad tutvuda uute andurite ning nende võimalustega ja hiljem jagada uusi teadmisi oma õpilastega. Tudengite loodud ülesanded abistavad õpetajaid, sest need on loodud erinevate raskusastmetega ja neid saab kasutada muutmata kujul koheselt tundides.

Tudengite loodud materjalidest pannakse kokku ka e-õppe kursused robotikast huvitatud üliõpilastele ja õpetajatele.

## **1.2. Hetkel RAJU keskuse poolt kasutatavad infotehnoloogilised vahendid ja kogutud informatsioon**

Igal aastal kirjutavad TÜ tudengid oma bakalaureusetöö raames uusi materjale LEGO MINDSTORMS robotitele mõeldud anduritest. Hiljem redigeeritakse need materjalid e-õppe kursuseks, mis asub hetkel Tartu Ülikooli Moodle keskkonnas nime all “LEGO MINDSTORMS NXT algkursus (MTAT.03.257)”. Sellel kursusel õpivad valdavalt

robotikahuvilised tudengid. Teine materjalide kasutussuund on kooliõpetajatele õppematerjalide valmistamine.

Kuni 2010. aastani oli materjalide redigeerimine eriti keeruline, sest puudus ühtne reeglistik, kuidas tuleks lõputöid vormistada nii, et hilisem redigeerimine oleks võimalikult kerge ja annaks ühtse terviku. 2010. aastal lahendas Jaana Metsamaa selle probleemi, koostades oma magistritööna materjalide kooskirjutamise raamistiku RAJU keskuse jaoks [7].

Materjalide ühtse vormi probleem sai küll lahendatud, aga puudub koht, kus saab neid materjale mugavalt hoida ja jagada. Hetkel olemasolevas kursuses saab materjale küll jagada, kuid puudub mugav liides nende haldamiseks. Aktiivsemad õpetajad on ka ise materjale ja töölehti loonud ja üldisele arengule oleks väga positiivne, kui neid materjale saaks samuti avalikustada, kahjuks hetkel sellist võimalust pole. RAJU keskuse paari aastase töö jooksul on valminud kümneid töölehti, juhendeid ja näidisülesandeid. Ka sel aastal uurivad ligi kakskümmend tudengit oma lõputööde raames LEGO MINDSTORMS robotitega ühilduvaid andureid, mille kohta puuduvad veel eestikeelsed kirjeldused ja juhendid. Nendest lõputöödest valmistatakse õpetajatele ja õpilastele lugemiseks sobivad õppematerjalid. Seega võib väita, et materjalide valik on tohutu.

Hetkel jagatakse materjale eelpool mainitud Moodle kursuses. Muutmata kujul tudengite lõputöödega on võimalik tutvuda RAJU keskuse koduleheküljel, mis asub aadressil <http://raju.cs.ut.ee/>. Samuti viiakse läbi koolitusi õpetajatele, kus selgitatakse LEGO MINDSTORMS robotite rakendusvõimalusi koolides ja kirjeldatakse erinevaid andureid, mida on võimalik robotikomplektide külge ühendada. Need koolitused on koostatud materjalide põhjal, mis on loodud üliõpilaste lõputööde baasil. Materjalide all mõeldakse siin ja edaspidi roboteid ja andureid kirjeldavaid õppetekste, ülesandeid ja õpilastele mõeldud töölehti. Soovi korral on õpetajatel võimalus saada materjale ka individuaalselt näiteks e-posti teel. Sellega aga praegu materjalide jagamise võimalused piirduvadki.

Õpetajatel puudub võimalus näha kõiki olemasolevaid materjale korraga ja neid soovi korral sirvida. See oleks aga vajalik juhul, kui näiteks kool saaks tellida uusi andureid, aga õpetaja

pole teadlik, missugused andurid üldse eksisteerivad ja millised on nende võimalused. Keskse infosüsteemi puhul, mis sisaldaks kõiki LEGO MINDSTORMS robotite eestikeelseid materjale, sellist probleemi ei tekiks, sest materjalid oleksid alati kõigile kättesaadavad ja kergesti otsitavad. Samuti kaoks küsimus - kus on võimalik leida uusi materjale? Ka RAJU keskusele oleks sellise süsteemi olemasolu kasulik - see võimaldaks saada kiiret ülevaadet olemasolevatest materjalidest ja õpetajatest.

Praegu on õpetajatel võimalus jagada töölehti ja ülesandeid oma õpilastele eKooli, mõne muu andmevahetussüsteemi kaudu või füüsiliselt paberikandjal. Õpetajate endi poolt koostatud ülesanded ja võimalikud seletused ja lisamaterjalid jäävad teistele õpetajatele kättesaamatuks. Keskne infosüsteem lahendaks ka selle probleemi - nii õpetajatel kui ka õpilastel oleks võimalik registreerida end selles infosüsteemis ja muretult jagada materjale. Soovi korral saaks teha materjalid avalikuks, mis võimaldaks neid näha ka teistel õpetajatel. Tulemusena valmiks kvaliteetsemad materjalid ja ülesanded ning tundide ettevalmistamine võtaks õpetajal vähem aega.

Seoses nende eelistega otsustas RAJU keskus luua süsteemi, mille täpsema kirjeldusega on võimalik tutvuda järgnevides peatükkides.

## 2. Uue infosüsteemi analüüs

Selle peatüki eesmärgiks on uue infosüsteemi täpsem uurimine ja analüüs: teha kindlaks, mis nõuded on süsteemile esitatud, millised on süsteemi komponendid ja kuidas need omavahel suhtlevad. Samuti peab uurima võimalike riske, mis tekivad seoses süsteemi kasutuselevõtmisega ja arendamisega.

### 2.1. Kasutajad

Infosüsteemil peaks olema 5 erinevat kasutajatüüpi. Igal tüübil on omad õigused ja võimalused.

1. **Anonüümne kasutaja** saab kasutada materjalide otsingut ja vaadata avalikke materjale, millel puudub piirang nähtavusele. Samuti võib registreerida endale uue kasutajaprofiili, mis on registreerimisjärgselt automaatselt õpilase rollis.
2. **Õpilane** saab lisaks anonüümse kasutaja võimalustele märkida materjalid lemmikuteks või lahendatuks. Mõlemal juhul satuvad materjalid eraldi tabelisse, kust on neid hiljem kergem üles leida ja teistest materjalidest eristada. Samuti saavad õpilased näha materjale, millel on piiratud nähtavus, aga mille autor on lisanud õpilase lubatud kasutajate nimekirja.
3. **Õpetaja** omab kõiki õpilase õigusi. Peale nende saavad õpetajad lisada infosüsteemi uusi materjale ja otsustada, kes neid näha võivad. Õpetajad saavad lisada teisi infosüsteemi kasutajaid endaga seotud kasutajate nimekirja, see on vajalik materjali vaatamise õiguste jagamisel. Protsessi lihtsustamiseks saab luua grupe, panna neile sobiv nimi ja paigutada õpilased gruppidesse. Samuti võib gruppi panna teise grupi, mille tagajärjel saavad kõik teise grupi liikmed esimese grupi õigused. Seejuures on kõik ühendused dünaamilised, mis tähendab seda, et isegi kui on kasutuses pikad gruppide jadad, siis kasutaja eemaldamisel grupist kaovad tal automaatselt ka kõik õigused, mis on antud grupile, mitte kasutajale endale.



4. **Moderaatoril** on lisaks kõigile õpetaja võimalustele, õigus kinnitada materjalid ja muuta need avalikuks. Uue materjali loomisel, ei muutu see koheselt kõigile avalikuks. Esmalt peab moderaator või administraator materjali üle vaatama ja veenduma, et see ei sisalda midagi ebasobivat. Seejärel saab ta kinnitada dokumendi korrektsuse. Alles peale seda saavad õpilased ja teised kasutajad materjali näha. Juhul, kui materjal on avalikult infosüsteemis üleval ja autor teeb mõne muudatuse, siis teised kasutajad seda muudatust esialgu ei näe, neile esitatakse vana materjali koopia. Moderaator või administraator peab materjali uuesti läbi vaatama ja ära kinnitama, alles peale seda saavad muudatust näha ka tavakasutajad. Lisaks kinnitamise õigusele, saavad moderaatorid muuta kõiki süsteemis olevaid materjale.
  
5. **Administraator** on suurimate õigustega kasutajatüüp. Lisaks kõigile moderaatori võimalustele, saab administraator hallata kõiki kasutajaid - muuta nende andmeid või hoopis süsteemist ära kustutada. Samuti saavad administraatorid muuta iga kasutaja tüüpi. Seda on tarvis siis, kui õpetajad registreerivad end süsteemis, siis alguses on konto tüübiks automaatselt õpilane. Selleks, et õpetajad saaksid lisada oma materjale, tuleb nende kontotüübiks panna õpetaja roll.

Moderaatori ja administraatori rollid on tähtsad seepärast, et RAJU keskus ei saa mingisugust tulu ja täiskohaga andmehalduri palkamine on mõeldamatu. Seetõttu tuleb mõnele aktiivsemale õpetajale ja RAJU keskuse liikmele anda õigus ja kohustus teha lisaks administratiivset tööd. Seda arvestades, peavad administreerimise lahendused olema võimalikud lihtsad ja mugavad, et haldajaid mitte liigselt koormata.

Alates õpilase kasutajast peavad kõik kasutajatüübid infosüsteemi sisse logima ja sellega kaasnevad parooli vahetamise ja meeldetuletamise funktsioonid.

## **2.2. Nõuded**

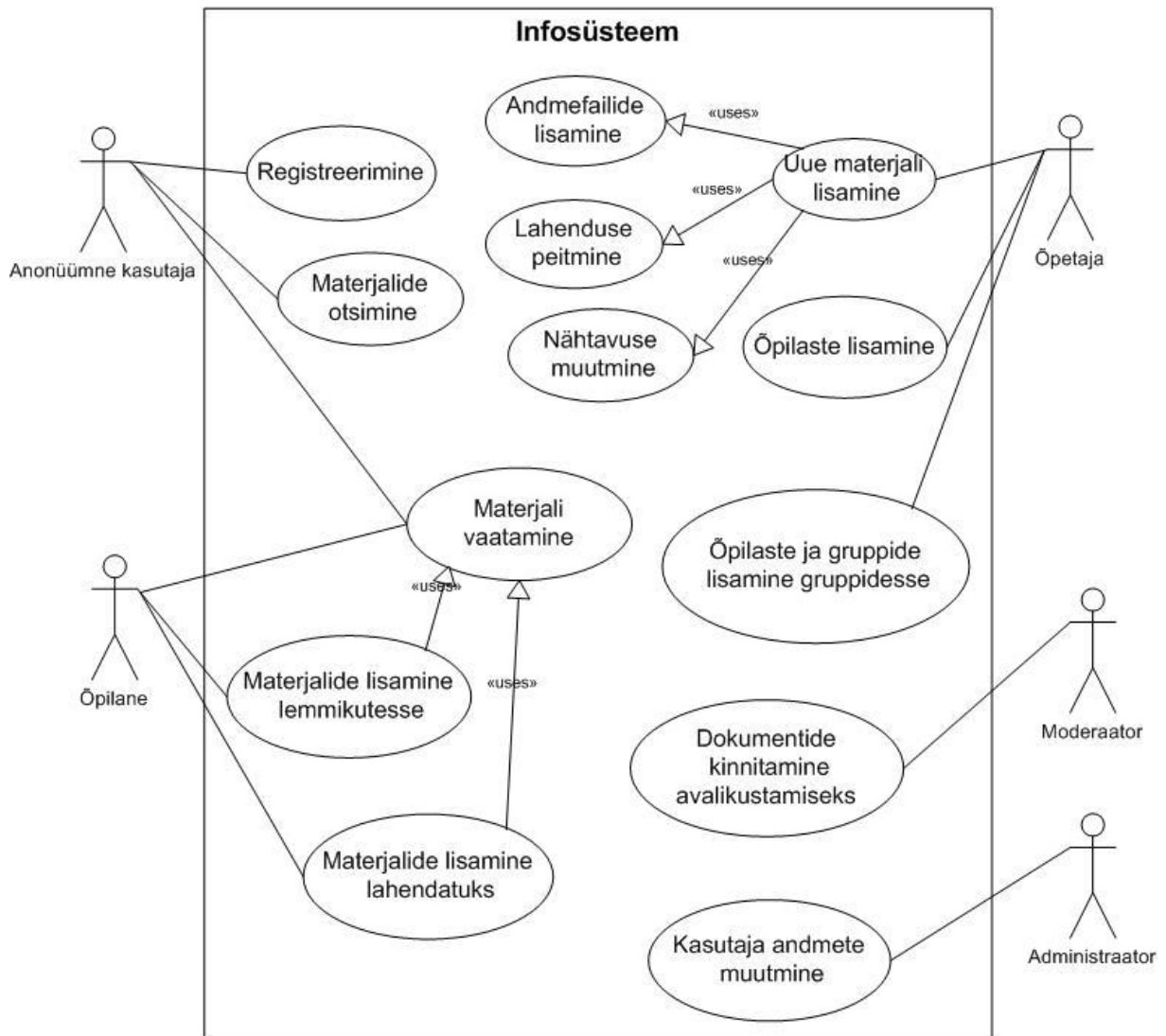
Nõuded jagunevad funktsionaalseteks ja mittefunktsionaalseteks nõueteks. Funktsionaalsed nõuded kirjeldavad süsteemi ja selle komponentide ülesandeid või probleeme, mida süsteem peab lahendama. Näidatakse, kuidas sisendist jõutakse samm-haaval väljundini [8]. Mittefunktsionaalsed nõuded pigem hindavad süsteemi, kui kirjeldavad selle käitumist. Kui funktsionaalsed nõuded kirjeldavad, mida süsteem peab tegema, siis mittefunktsionaalsed nõuded kirjeldavad, milline peaks süsteem olema [9]. Funktsionaalsed nõuded on esitatud antud peatükis kasutuslugudena.

Nõuded saadi RAJU keskuse eestvedajate Anne Villemsi ja Taavi Duvini käest nõuetedokumendi näol, mis on lisatud ka antud töö lisade alla (lisa 1).

### **2.2.1. Kasutuslood**

Kasutuslood on konkreetsete sammude kirjeldused, mis tuleb kasutajal täita, et jõuda mõne kasutaja poolt püstitatud eesmärgini. Kasutaja all võib mõelda nii infosüsteemi kasutavat füüsilist isikut, kui ka midagi abstraktsemat, näiteks väline süsteem või protsess.

Järgneval joonisel on kujutatud kõik süsteemi kasutajad ja nende põhilised tegevused.



Joonis 1. Kasutuslugude diagramm. <uses> nool tähistab seda, et kasutuslugu kasutab mingit teist kasutuslugu.

## **Kasutuslugu nr. 1 - Registreerimine**

**Olukorra kirjeldus:** Kasutaja registreerib endale uue kasutajakonto

**Eeltingimused:** Kasutajal on juurdepääs Interneti. Avatud on infosüsteemi leht.

**Oodatav tulemus:** uus kasutajakonto on registreeritud ja kasutaja sisselogitud

**Kohustuslikud parameetrid:** eesnimi, perenimi, kasutajanimi, email, salasõna, pildi sümbolid

**Valikulised parameetrid:** -

**Põhivoog:**

1. Kasutaja valib menüüst "Registreeri" lehe.
2. Kasutaja sisestab oma andmed
3. Süsteem suunab kasutaja vastloodud konto esilehele.

**Alternatiivvoog:** -

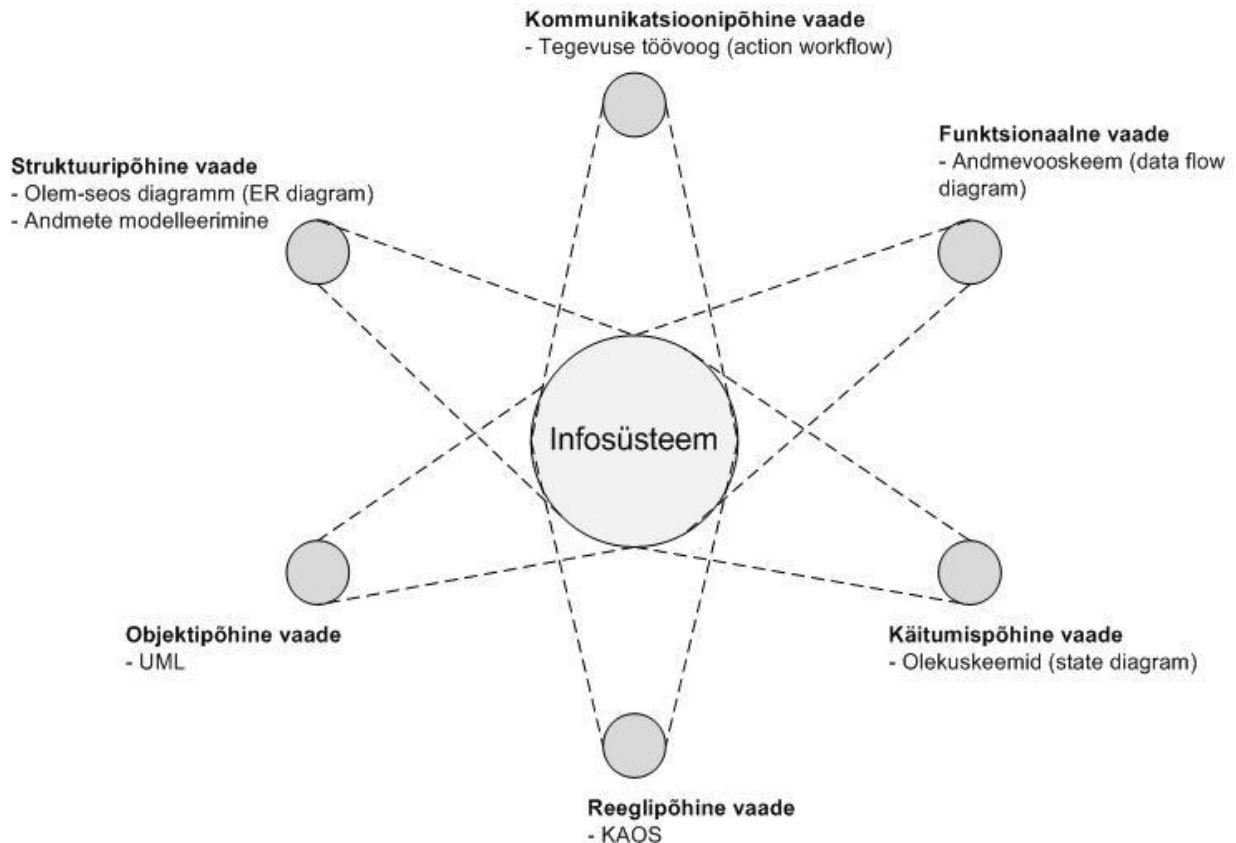
**Tulemus veasituatsiooni korral:** Kasutajale kuvatakse vastavad veateated ja suunatakse punkti 2.

**Märkused:** Õpetaja poolt antud lingiga (kätte saadav "Õpilased" lehelt) registreerimisel seotakse kasutajakonto automaatselt õpetaja omaga.

Ülejäänud kasutuslood on paigutatud lissasse 2.

## **2.3. Struktuur**

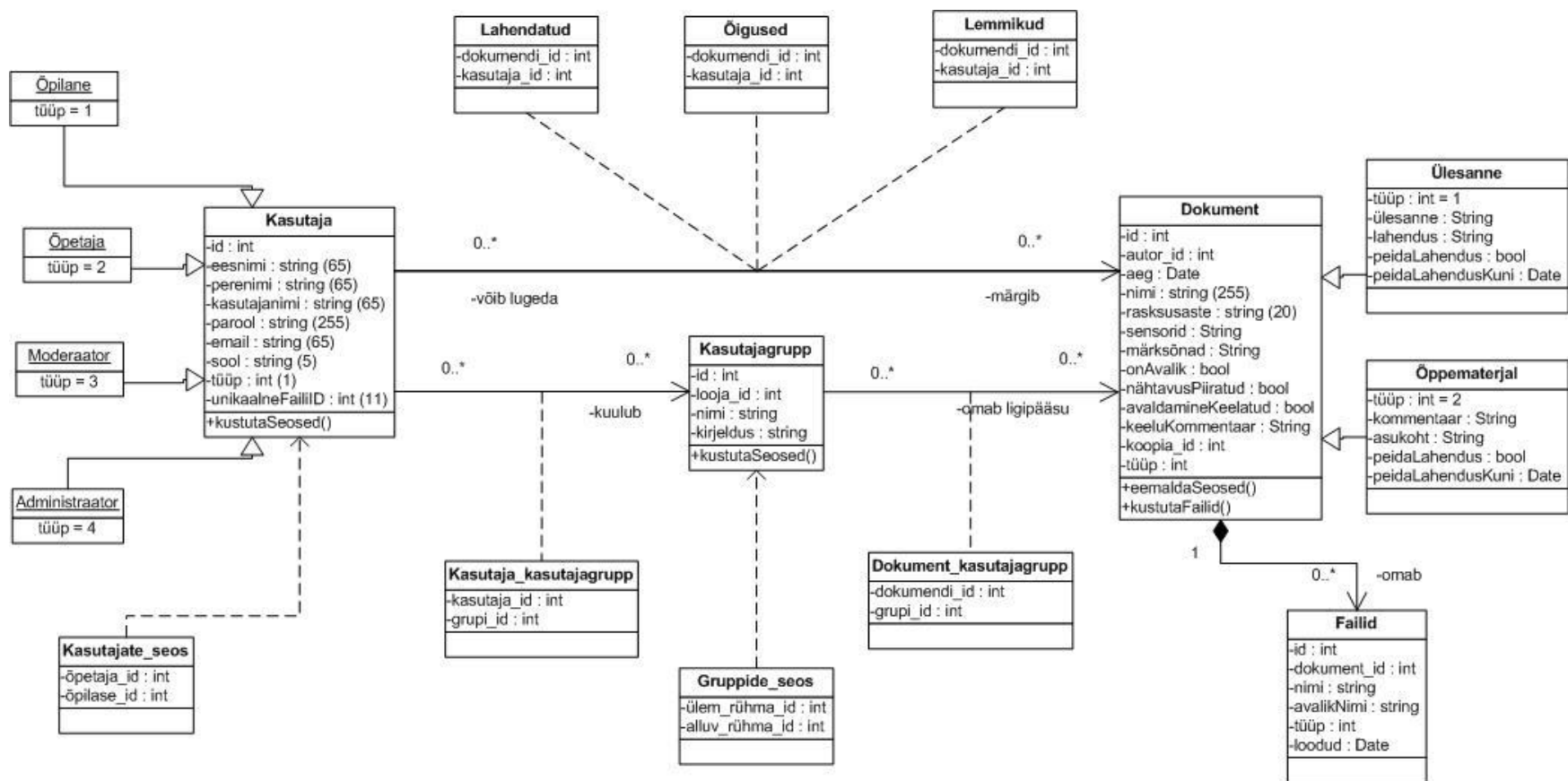
Süsteemi analüüsi on võimalik jagada funktsionaalseks, struktuuri-, käitumis-, reegli-, objekti- ja kommunikatsioonipõhiseks analüüsiks [10]. Need analüüsi vaated täiustavad üksteist - ühe vaatega on võimalik analüüsida selliseid süsteemi osi, mida teine ei kata. Seda väidet illustreerib joonis 2.



Joonis 2. Infosüsteemi analüüsi vaated. Joonise loomisel on kasutatud infosüsteemi erinevaid vaateid tutvustavaid allikaid [10, 11].

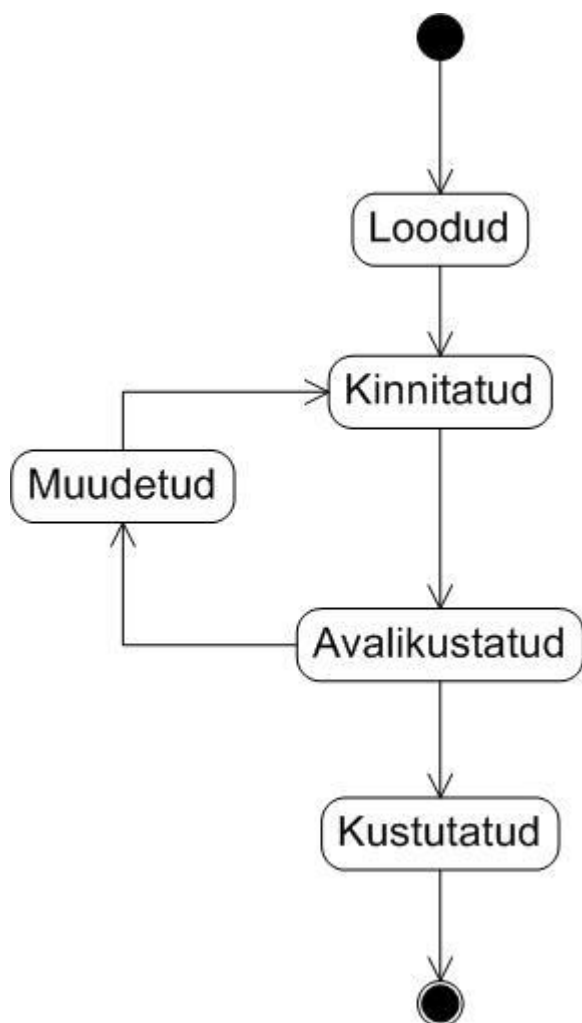
Antud töös on valitud käitumis- ja objektipõhine analüüs, sest uurivad süsteemi erinevate vaatenurkade alt ja kokku annavad küllaltki ülevaatliku pildi arendatavast süsteemist. Objektipõhise analüüsi osa täidab joonis 2 ja käitumispõhise vaadet kujutab joonis 3.

Joonisel 3 on kujutatud klassidiagrammi, mis kirjeldab süsteemi struktuuri, näidates klasse, nende atribuute ja meetodeid ning seoseid klasside vahel [12]. Klassidiagramm on üks osa UML (*Unified Modeling Language*) keelest, mis on standardiseeritud objekt-orienteeritud tarkvaraarenduse modelleerimiskeel [13].



Joonis 3. Klassidiagramm

Kuna kogu süsteem tiirleb ümber materjalide, siis on mõistlik näidata materjalide erinevaid võimalikke olekuid, mis süsteemis olla võivad. Selleks on loodud joonis 4.



Joonis 4. Materjali olekud

Igal materjalil on kohustuslik nimi. Valikuliselt saab lisada ka märksõnu ja materjalis kirjeldatud andureid. Kõik materjalid on jagatud kolmeks raskustasemeks. Materjale on võimalik otsida või sorteerida kõigi eelpool nimetatud suuruste järgi. Kõigile materjalidele on võimalik lisada ka üks või enam faili (ülesannete korral nii ülesande kui lahenduse juures eraldi). Failide kogusele ja mahule piiranguid pole seatud, kuid seda on võimalik lihtsasti lisada. Õppematerjalide puhul loetakse üleslaetud zip formaadis fail veebileheks ja üritatakse

seada kasutajale kuvada. Kui tegemist on ülesandega, siis lahendust on võimalik peita valitud kuupäevani.

Kui materjal on loodud, siis on ta nähtav ainult autorile ja moderaatoritele ning administraatoritele, kes peavad selle materjali esmalt kinnitama, enne kui õpilased seda näha saavad. Peale dokumendi kinnitamist läheb see automaatselt avalikustatud olekusse ja vastavalt seadetele näevad seda kõik või ainult valitud kasutajad. Kui avalikustatud dokumenti muuta, siis sellest tehakse koopia, mis jääb kõigile nähtavaks. Muudetud materjal vajab uuesti moderaatori või administraatori kinnitust. Peale kinnituse saamist koopia kustutatakse ja kasutajad näevad materjali juba muudetud kujul.

## 2.4. Riskid

Ükski projekt pole täiesti riskivaba [14]. Seepärast tuleks riske võimalikult varakult tuvastada ja mõelda selle peale, kuidas nende poolt tekitatav kahju võimalikult efektiivselt maandada. Järgnevas tabelis on välja toodud antud süsteemi arendamisega kaasnevad suurimad riskid, nende olulisus ja ilmnemise tõenäosus.

<b>Risk</b>	<b>Tähtsus</b>	<b>Ilmnemise tõenäosus</b>
Kirjeldatud süsteemi programmeerimine osutub liialt keerukaks.	Suur	Keskmine
Süsteemi nõuded osutuvad vastuolulisteks	Suur	Väike
Süsteem osutub mahukamaks ja tekib ajapuudus	Suur	Keskmine
Süsteemi komponendid osutuvad omavahel mitteühilduvateks	Keskmine	Väike
Süsteemi disain ei meeldi kasutajatele	Väike	Keskmine

Tabel 1. Riskid

Kõikide riskidega on arvestatud ja probleemide tekkimisel on võimalik need lahendada konsulteerides RAJU keskusega ja muutes süsteemi nõuded.



### 3. Loodud süsteemi kirjeldus

Veebipõhise infosüsteemi arendamisel on tehnoloogiate ja keelte valik väga suur. Serveri poole pealt saab valida näiteks ASP, ColdFusion, Perl, Groovy, Java, PHP, Python, Ruby, Smalltalk, SSJS ja .NET keelte vahel, kliendi poole pealt JavaScript, Flash, Microsoft Silverlight HTML [15]. Ka andmebaase on mitmeid: Apache Derby, DB2, Firebird, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, SQLite ja Sybase. Kõigil neist on omad eelised ja puudused ning pole võimalik öelda, et üks on teisest parem.

Lõppvalik peatus serveri poole pealt PHP skriptimiskeelel, kliendipoolse keelena kasutatakse valdavalt JavaScripti (uue õppematerjali üleslaadimine toimub Flashi abil, sest JavaScript ei toeta failide üleslaadimist) ning andmebaasiks on MySQL. PHP keele valiku põhjuseks on see, et PHP ei vaja eraldi mingisugust virtuaalmasinat ja töötab kõigil platvormidel ilma vajaduseta koodi muuta. Samuti on see väga populaarne ja Internetis leidub väga palju PHP alast õppematerjali ja koodinäiteid. JavaScript ei nõua kasutajalt mingisuguse lisatarkvara paigaldamist ja on automaatselt toetatud suuremate veebilehitsejate poolt. See saigi JavaScripti valiku põhjuseks. MySQL andmebaasil ei olnud selgeid eeliseid teiste valikute ees, kuid selle paigaldamise lihtsus ning eelnev kogemus antud andmebaasiga kallutasid lõppotsuse MySQL andmebaasi poole.

#### 3.1 Arhitektuur

Programmeerimise lihtsustamise ja koodi loetavuse huvides on valitud mudel-vaade-kontroller (*MVC - model-view-controller*) struktuur [16]. Teised võimalikud struktuuritüübid on näiteks:

- Lehe kontroller – peakontrolleril on alamkontrollerid iga loogilise lehe (päringu) jaoks. Alamkontroller võib olla HTML leht ise või mõni keerulisem objekt, milles on päringu töötlemise loogika.
- Eeskontroller – keerulisemates süsteemides on tihti asju, mida tuleb teha iga päringu ajal, näiteks turvalisuse kontroll. Vältimaks koodi kopeerimist mitmesse erinevasse

kontrollerisse, saadetakse päringud eeskontrollerile, mis töötleb päringu ära ja seejärel saadab selle edasi alamkontrollerile.

- Mall – dünaamiliste päringute jaoks koostada staatilised lehed, kuhu saab lisada dünaamiliselt infot.

Ülejäänud struktuuridega saab tutvuda veebilehel [16].

Mudel-vaade-kontroller struktuur on kõige sobivam lahendus, et hoida kood võimalikult lihtsana ja samas vähendada sama koodi kordamist erinevates süsteemi osades. Mudel-vaade-kontroller struktuuri puhul asub kogu info töötlemine ja arvutamine kontrolleris [17]. Andmebaasiga otseselt ei suhelda, vaid kasutatakse selleks mudelit, mis tegutseb vahendajana kontrolleri ja reaalse andmebaasi vahel. Samuti puudub kontrolleris kasutajaliidese kujunduse kood. Töödeldud andmed antakse edasi vaatele, mis paigutab need vastavalt eeldefineeritud kujundusele sobivaks veebileheks. Kui saabub uus päring, siis suunatakse see vastavale kontrollerile ning kontroller kutsub välja vastavalt vajadusele mudeli ja nõutava vaate.

Et lihtsustada programmeerimist veelgi ja lisada loodavale süsteemile stabiilsust, on otsustatud kasutada programmeerimisel CodeIgniter raamistikku [18]. Tegemist on vabavaralise, väikese, kiire ja selge raamistikuga, mis ei vaja praktiliselt mingisugust seadistamist ja pole tarvis kasutada käsurida, mida kasutavad paljud konkureerivad raamistikud. Samuti on raamistik hästi dokumenteeritud ja omab küllaltki suurt ning abivalmit kogukonda. Raamistiku kasutamine kiirendab programmeerimist, sest arendajale on ette tehtud hulk funktsioone, mida saab koheselt kasutada ja mida pole vaja endal uuesti teostada. Näiteks on olemas sessioonihaldus, reaalne andmebaasiga ühildumine ja andmete pärimine toimub tagataustal ja programmeerija neid päringuid ei näe. Stabiilsuse tagab eelseadistatud kasutaja sisendite kontroll ja valideerimine. Lisaks on loodud hulk abiklasse, mille abil on võimalik luua nõutud funktsionaalsus kordades lühema koodiga. Täispika CodeIgniter raamistiku kirjeldusega on võimalik tutvuda aadressil [19].

## 3.2. Kasutajaliides

Kasutajaliides on loodud HTML elementide abil ja dünaamilisuse lisab JavaScript. JavaScripti toetamiseks kasutatakse jQuery teeki [20], millel on hulk eeldefineeritud funktsioone ja mille abil saab programmeerimise kiirust tunduvalt kiirendada.

Eriliste kasutajaliidese elementide jaoks, näiteks vormi sammudeks jagamine, mitme faili korraga üleslaadimine ja kuupäeva valimine, on loodud eraldi jQuery teegil baseeruvad pistikprogrammid (*plugin*). Tabelis 2 on nimetatud kõik hetkel kasutuses olevad pistikprogrammid ja nende koduleheküljed.

Nimi	Kirjeldus	Kodulehekülg
Uploadify	Failide üleslaadimine	<a href="http://www.uploadify.com/">http://www.uploadify.com/</a>
jQuery Formwizard	Vormide jagamine sammudeks	<a href="http://thecodemine.org/">http://thecodemine.org/</a>
jQuery Multiple File Upload	Failide üleslaadimine	<a href="http://www.fynetworks.com/jquery/multiple-file-upload/">http://www.fynetworks.com/jquery/multiple-file-upload/</a>
jQuery Timepicker Addon	Aja/kuupäeva valimine	<a href="http://trentrichardson.com/examples/timepicker/">http://trentrichardson.com/examples/timepicker/</a>
jGrowl	Teadete tekitamine	<a href="http://stanlemon.net/projects/jgrowl.html">http://stanlemon.net/projects/jgrowl.html</a>

Tabel 2. jQuery pistikprogrammid

Lisaks jQuery teegile on pikemates tekstiväljades, näiteks ülesannete ja õppematerjalide koostamisel ülesande, lahenduse ja kommentaari väljad, teksti disaini ja kuva muutmise võimaldamiseks kasutatud CKEditor nimelist tekstiredaktorit. CKEditor on vabavaraline avatud lähtekoodiga tekstiredaktor, mis on mõeldud kasutamaks veebilehtedel [21]. Programmeeritud on see JavaScript keeles ja testitud kõigil suurematel veebilehitsejatel.

### 3.3. Suhtlus andmebaasiga

Andmebaasiga suhtlemiseks kasutatakse spetsiaalselt CodeIgniter raamistikule programmeeritud pistikprogrammi Datamapper [22], mis seob andmebaasis olevad tabelid objektideks ja võimaldab sellisel viisil andmeid vähema vaevaga pärida ja salvestada.

Andmete küsimisel kasutatakse *active record* struktuuri - andmebaasi tabelist tehakse klass, mis teab seoseid teiste tabelitega ja mille muutujad on tabeli väärtusteks. Näiteks andmebaasis on olemas tabel *Users*, mis sisaldab infot süsteemi kasutajate kohta. Sellel tabelil on väljad *username* ja *password*. *Active record* puhul näeks uue kasutaja lisamine välja järgmiselt:

```
user = new User()  
user.username = "DemoUser"  
user.password = "DemoPassword"  
user.save()
```

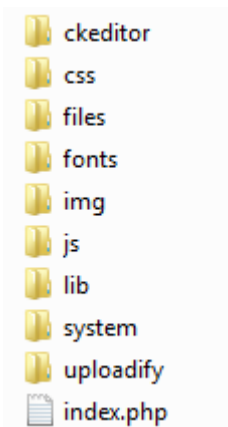
Vastavalt sellele koodile genereeritakse taustal reaalne päring:

```
INSERT INTO users (username, password) VALUES ('DemoUser',  
'DemoPassword');
```

Tegelikud teostused võivad erinevates raamistikutes erineda, kuid põhimõte jääb alati samaks [23]. Reeglina lisatakse ka mitmeid funktsioone andmete pärimiseks, kopeerimiseks ja valideerimiseks.

### 3.4. Süsteemi installeerimine

CodeIgniter raamistik ei vaja mingeid eelnevaid seadistusi, piisab vaid failide kopeerimisest serverile. Serverilt eeldatakse, et sellel on töökorras veebiserver, mis toetab PHP versioon 5. Selle lõputööga on kaasas fail *susteem.zip* (lisa 3). See tuleks lahti pakkida ja kopeerida serverisse sobivasse kataloogi. Süsteemi kataloogi struktuur on märgitud joonisel 5.



Joonis 5. Süsteemi kataloogi struktuur.

ckeditor, uploadify ja lib kaustad on spetsiaalsed JavaScript pistikprogrammide failid, fonts kaust sisaldab kirjastiile, mida kasutab Captcha (registreerimisel pildil olevate sümbolite küsimine kasutajalt, kontrollimaks, et tegemist on inimesega, mitte automaatse skriptiga). css kaust sisaldab süsteemi kujunduse faile, js JavaScripti faile ning img pilte. Kaustas files hoiustatakse kõiki süsteemi materjale. Kopeerimisel tuleks veenduda, et sellele kaustale ja selle alamkaustadele on määratud kirjutamisõigused. Lõpuks kaust system on CodeIgniter raamistik ise ja süsteemi lähtekood.

Nüüd peaks raamistik olema edukalt paigaldatud ja töökorras, kuid veel ei leita õiget teekonda kontrolleriini ja ei teata, mis andmebaasi kasutada. Selleks tuleb muuta kahte faili.

Kõigepealt tuleb avada fail `config.php`, mis asub `/system/application/config/` kaustas. Tuleb muuta rida 14 ja kirjutada sinna süsteemi uus aadress:

```
$config['base_url'] = "http://localhost/WebSites/Raju/";
```

Seejärel tuleb avada `database.php` fail, mis asub samuti `/system/application/config/` kaustas. Muuta tuleks ridu 40 - 45, vastavalt andmebaasi seadmetele:

```
$db['default']['hostname'] = "localhost";  
$db['default']['username'] = "root";  
$db['default']['password'] = "";  
$db['default']['database'] = "raju";  
$db['default']['dbdriver'] = "mysql";  
$db['default']['dbprefix'] = "";
```

Andmebaas ise tuleb eelnevalt ise valmis teha. Tabelite loomiseks tuleks kasutada selle tööga kaasas olevat `database.sql` faili, mis tekitab automaatselt kõik vajaminevad tabelid ja seosed nende vahel.

Sellega peaks süsteem olema edukalt installeeritud ja valmis kasutamiseks. Algselt on süsteemis ainult üks administraatori õigustega kasutaja, kelle kasutajatunnuseks ja parooliks on *admin*. Selle kasutajaga on võimalik tekitada reaalseid administraatoreid ja moderaatoreid. Peale esialgset seadistust on soovitatav see kasutaja eemaldada või muuta selle kasutaja parool.

## Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua infosüsteem, mis haldaks RAJU keskuse poolt loodud LEGO MINDSTORMS NXT robotite ja neile mõeldud andurite õppematerjale ning töölehti. Materjalide loomine ja jagamine peaks olema võimalikult lihtne. Infosüsteemi kasutajateks on Kooliroboti projektiga liitunud koolid. Õpetaja staatuses kasutajad saavad tutvuda uute andurite ja nende võimalustega. Samuti saavad nad koostada uusi ülesandeid, mida on võimalik mugavalt jagada oma õpilastele.

Lõputöö tekstiline osa esitab infosüsteemi analüüsi, kirjeldades erinevaid nõudeid, struktuuri ja ehitust. Viimane peatükk sisaldab süsteemi paigaldamisjuhendit. Süsteem ise on arendatud PHP keeles ja arendatud vabavaralise, väikse ja kiire CodeIgniter MVC (mudel-vaade-kontroller) raamistiku peale. Kasutajaliidese mugavamaks muutmiseks on kasutatud valdavalt JavaScripti teeki jQuery, vaid õppematerjalide üleslaadimisel tuli kasutada Uploadify pistikprogrammi, mis baseerub osaliselt Flash peal.

Süsteem on sisuliselt valmis kasutamiseks, kuid enne selle kasutuselevõttu oleks kasulik lasta mõnel disaineril luua süsteemile sobiv kujundus, mis vastaks lehe sisule ja jätaks professionaalse mulje.

Eelnevalt polnud sarnast süsteemi kasutuses ja programmeerimist tuli alustada tühjalt kohalt. Samuti osutus mitmete pistikprogrammide integreerimine antud raamistikuga ootamatult keeruliseks, mistõttu tuli teha mitmeid kompromisse. Põhifunktsionaalsus on töökorras, kuid kahjuks jäi aega puudu põhjalikuks testimiseks. Seepärast soovitaks enne päris kasutajate lisamist läbi teha ka korralik testimise tsükkel.

Järgmise sammuna tuleks tutvustada infosüsteemi koolidele ja suunata neid seda kasutama. Mida rohkem on süsteemis kasutajaid, seda rohkem loovad nad uusi materjale ja töölehti, mida teised kasutajad saavad uurida. Sellega aga tõuseb loodetavasti üldine robotika tase ja populaarsus.

# **Document management system for RAJU keskus**

Bachelor Thesis

Martin Suvorov

## **Summary**

Koolirobot project started in 2007. Its objective was to promote interest in engineering and start according activities in Estonian schools. To support this objective, the project helped schools buy LEGO MINDSTORMS NXT robotic kits and start robotics courses. RAJU keskus (Robotic Exploration and Development Centre) is an informal organisation at the Institute of Computer Science at the Tartu University. In this centre students create study materials and worksheets for LEGO MINDSTORMS robotic kits.

During the last three years the amount of materials has grown dramatically, but there is no convenient way to manage and share them with teachers and students. Therefore it was decided that a new document management system had to be created. The aim of this bachelor thesis is to create an information system to manage robotics study materials.

The textual part of this thesis serves as an analysis for the created system. In general this document can be divided into three parts - the background, analysis and overview.

The first chapter gives an overview about RAJU keskus and its current capabilities in the field of IT. It is explained why the new system is necessary and what is wrong with the current situation.

The second chapter describes system requirements, future user types and their capabilities. The overall system structure is specified.

The last chapter describes how the system was created. Here the reader can learn that the system was written in PHP language on server side. CodeIgniter was chosen as MVC



framework for the system. For better user experience jQuery JavaScript library was used. At the end of the third chapter an installation guide is included.

The system is ready for installation and examination. It is recommended that before going public with this system, a new design would be created by a professional designer.

The created system is included on CD at the end of this document.

## Viited

1. Wikipedia, Robot <http://et.wikipedia.org/wiki/Robot> (viimati vaadatud 21.02.2011)
2. Wikipedia, Robotika <http://et.wikipedia.org/wiki/Robotika> (viimati vaadatud 21.02.2011)
3. Mehhatroonikainstituut <http://www.ttu.ee/mehhatroonikainstituut/avaleht/> (viimati vaadatud 26.02.2011)
4. MTÜ TTÜ Robotiklubi "TTÜ Robotikaklubi" <http://www.robotiklubi.ee/avaleht> (viimati vaadatud 21.02.2011)
5. ITK Robotikaklubi <http://robot.itcollege.ee/?q=node/1> (viimati vaadatud 21.02.2011)
6. MTÜ Robotika "Kooliroboti komplekt" <http://www.robootika.ee/lego/projekt/index.php/projektist/> (viimati vaadatud 21.02.2011)
7. Jaana Metsamaa „Materjalide kooskirjutamise raamistik RAJU keskuse õppevara näitel“ 2010 <http://raju.cs.ut.ee/loputood/2010/jaana-metsamaa/> (viimati vaadatud 17.05.2011)
8. Wikipedia, Functional requirement [http://en.wikipedia.org/wiki/Functional\\_requirement](http://en.wikipedia.org/wiki/Functional_requirement) (viimati vaadatud 06.03.2011)
9. Wikipedia, Non-functional requirement [http://en.wikipedia.org/wiki/Non-functional\\_requirement](http://en.wikipedia.org/wiki/Non-functional_requirement) (viimati vaadatud 03.02.2011)
10. Raimundas Matulevičius "Information systems modelling methodology and perspectives" 27.09.2010 <http://courses.cs.ut.ee/2010/is/uploads/Main/03-IS-methodology-and-perspectives.pdf> (viimati vaadatud 17.05.2011)
11. Raimundas Matulevičius "Exercise 2: IS Modelling" 09.11.2010 <http://courses.cs.ut.ee/2010/is/uploads/Main/Exercise%202.pdf> (viimati vaadatud 17.05.2011)
12. Wikipedia, Class Diagram [http://en.wikipedia.org/wiki/Class\\_diagram](http://en.wikipedia.org/wiki/Class_diagram) (viimati vaadatud 03.05.2011)
13. Wikipedia Unified Modeling Language [http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language) (viimati vaadatud 14.05.2011)

14. Yen Yee Chong, Evelyn May Brown “Managing project risk: Business risk management for project leaders”, Inglismaa 2000
15. Wikipedia, Web development [http://en.wikipedia.org/wiki/Web\\_development](http://en.wikipedia.org/wiki/Web_development) (viimati vaadatud 07.04.2011)
16. Martin Fowler “Catalog of Patterns of Enterprise Application Architecture” 2003 <http://martinfowler.com/eaCatalog/> (viimati vaadatud 16.05.2011)
17. Tony Marston “The Model-View-Controller (MVC) Design Pattern for PHP” 26.06.2010 <http://www.tonymarston.net/php-mysql/model-view-controller.html> (viimati vaadatud 17.05.2011)
18. CodeIgniter <http://codeigniter.com/> (viimati vaadatud 10.04.2011)
19. CodeIgniter Features [http://codeigniter.com/user\\_guide/overview/features.html](http://codeigniter.com/user_guide/overview/features.html) (viimati vaadatud 15.05.2011)
20. jQuery <http://jquery.com/> (viimati vaadatud 11.04.2011)
21. CKEditor <http://ckeditor.com/> (viimati vaadatud 11.04.2011)
22. DataMapper <http://stensi.com/datamapper/> (viimati vaadatud 11.04.2011)
23. Wikipedia, Active Record Pattern [http://en.wikipedia.org/wiki/Active\\_record\\_pattern](http://en.wikipedia.org/wiki/Active_record_pattern) (viimati vaadatud 31.03.2011)

# Lisad

## Lisa 1. Süsteemi nõuded

### Üldinfo:

Portaal/süsteem, kus on õpilastele-õpetajatele erinevad materjalid, juhendid jne

### Nõuded:

- Lihtne kasutada ja hallata, toimiv kõigi op süsteemidega, ei tohiks kasutaja arvutilt nõuda väga suurt ressursi (serveri ressursidega pole ilmselt probleemi, aga ka neid ei tohiks raisata :))
- 4 kasutajate grupp: õpilane/tavakasutaja, õpetaja, moderaator, administraator
  - õpilase õigused: saab vaadata juhendeid/õpetusi/vastava loaga ülesandeid
  - õpetaja õigused: kõik õpilase omad+saab ülevaatomiseks lisada ülesandeid, võimalus näha ja kirjutada ülesannete juurde kommentaare
  - moderaator: õpetaja õigused + võimalus kinnitada ülesannete sobilikkus, õpetajate kasutajate loomise võimalus
  - administraator - moderaatorite loomise võimalus ja üldse kõige võimsam
- Ülesanded-õpetused peavad olema miskitmoodi grupeeritavad ja lihtsasti otsitavad (uusi grupe peaks saama juurde teha): nt
  - grupid raskusastme järgi: WeDo, algaja, kesktase, kõrgem kesktase, kõrgtase
  - sorteerimine kasutatava anduri järgi: nt ülesanded, kus on kasutuses ainult mootorid (peaks olema ka variant, et mootorid + nt valgusandur), ülesanded, kus on kasutuses ainult valgusandur (samuti variant valgusandur+miskit vel)
- ülesande lisamisel nõ Tag'de lisamise võimalus otsingu jaoks. nt kui ülesanne kasutab mootorit ja heliandurit siis ülesande juures on mäрге, et kasutatakse heliandurit ja mootorit
- Hästi tore oleks kui süsteem oleks umbes sellise kasutajaliidesega nagu Markit'i arvuti osade koduleht (<http://www.markit.eu/ee/et/koik-sulearvutid/v1c5>). Näitab ära kui palju on mingile raskusastmega ülesandeid vastava anduri kohta jne umbes midagi sellist: (neid ribasid nn tag'e võiks saada ka lisada)

Andur: valgusandur (7), mootorid(19), heliandur (8)

Raskusaste: WeDo(1), algaja (8), kesktase (16), kõrgem kesktase (1)

Lemmikud: Lahendatud (4), Lahendamata (20)

- Ülesande lisamine võiks olla samuti hästi lihtne, võiks sisaldada väljasid:
  - Ülesande osa
    - Ülesande nimi
    - Raskusaste
    - Märksõnad
    - Ülesande tekst
    - Selgitavad pildid ülesande juurde
    - Võimalus viidata sobivatele õpetustele/juhendmaterjalidele
  - Lahenduse osa
    - Tekkida võivad probleemid
    - Lahenduskäik
    - Lahenduse failid (näidisprogrammid)
- Õpetajatel peaks olema võimalus omale nõ lemmikülesandeid märkida ja nendest nimekirja koostada. Mingi variant, et märgitakse ära ülesanded, mis on juba lahendatud, välja valitud vms

## Lisa 2. Kasutuslood

### Kasutuslugu nr. 2 - Materjali otsimine

**Olukorra kirjeldus:** Kasutaja tahab otsida materjali

**Eeltingimused:** Kasutajal on juurdepääs Internetti. Avatud on infosüsteemi leht.

**Oodatav tulemus:** kasutajale kuvatakse nimekiri otsingutulemusega

**Kohustuslikud parameetrid:** -

**Valikulised parameetrid:** märksõna

**Põhivoog:**

1. Kasutaja valib menüüst "Otsing" lehe.
2. Kasutaja sisestab märksõna
3. Süsteem otsib materjale, mille nimes, autori nimes, kasutatavate andurite või märksõnade nimekirjas esineb sisestatud märksõna.
4. Kasutajale väljastatakse nimekiri otsingutulemusega.

**Alternatiivvoog:**

2. Kasutaja valib tabelist sobiva raskusastme või tihedamini esineva anduri või märksõna
3. Vastavalt valikule, otsitakse kõik materjalid valitud raskusastme, anduriga või märksõnaga.
4. Kasutajale väljastatakse nimekiri otsingutulemusega.

**Tulemus veasituatsiooni korral:** Kasutajale kuvatakse tühi tulemustetabel

**Märkused:** Piiratud nähtavusega materjale pole võimalik otsingust leida. Need on automaatselt olemas "Minu materjalid" lehel, kui kasutajal on vastav õigus materjali näha.

### Kasutuslugu nr. 3 - Materjali lisamine lemmikutesse

**Olukorra kirjeldus:** Kasutaja lisab materjali oma lemmikutesse

**Eeltingimused:** Kasutaja on süsteemi sisse loginud. Kasutajal on avatud materjal, mida ta soovib lemmikutesse lisada

**Oodatav tulemus:** materjal on edukalt lisatud lemmikutesse ja nähtav "Minu materjalid" lehel vastava märgiga.

**Kohustuslikud parameetrid:** -

**Valikulised parameetrid:** -

**Põhivoog:**

1. Kasutaja valib “Märgi lemmikuks” lingi.
2. Süsteem salvestab materjali lemmikute nimekirja ja kuvab seda “Minu materjalid” lehel vastava märgiga.
3. Materjali leht uuendatakse ja “Märgi lemmikuks” lingi asemel on “Eemalda lemmikutest” kiri.

**Alternatiivvoog:** -

**Tulemus veasituatsiooni korral:** Mingeid muudatusi ei tehta

**Märkused:** -

#### **Kasutuslugu nr. 4 - Materjali märkimine lahendatuks**

**Olukorra kirjeldus:** Kasutaja märgib materjali lahendatuks

**Eeltingimused:** Kasutaja on süsteemi sisse loginud. Kasutajal on avatud materjal, mida ta soovib lahendatuks märkida.

**Oodatav tulemus:** materjal on edukalt märgitud lahendatuks ja nähtav “Minu materjalid” lehel vastava märgiga.

**Kohustuslikud parameetrid:** -

**Valikulised parameetrid:** -

**Põhivoog:**

1. Kasutaja valib “Märgi lahendatuks” lingi.
2. Süsteem salvestab materjali lahendatud materjalide nimekirja ja kuvab seda “Minu materjalid” lehel vastava märgiga.
3. Materjali leht uuendatakse ja “Märgi lahendatuks” lingi asemel on “Eemalda lahendatud märk” kiri.

**Alternatiivvoog:** -

**Tulemus veasituatsiooni korral:** Mingeid muudatusi ei tehta

**Märkused:** -

## **Kasutuslugu nr. 5 - Materjali vaatamine**

**Olukorra kirjeldus:** Kasutaja vaatab materjali

**Eeltingimused:** Kasutajal on juurdepääs Internetti. Avatud on infosüsteemi leht.

**Oodatav tulemus:** kasutaja näeb soovitud materjali

**Kohustuslikud parameetrid:** -

**Valikulised parameetrid:** -

**Põhivoog:**

1. Kasutaja on otsingu lehel ja sooritanud eduka otsingu
2. Kasutaja valib tabelist soovitud materjali ja vajutab materjali nimele.
3. Süsteem kontrollib, kas kasutajal on õigus antud materjali vaadata.
4. Eduka kontrolli puhul kuvatakse kasutajale materjal

**Alternatiivvoog:**

1. Kasutaja on "Minu materjalid" lehel.
2. Kasutaja valib nimekirjast soovitud materjali ja vajutab materjali nimele.
3. Liigutakse punkti 3.

**Tulemus veasituatsiooni korral:** Kui kasutajal puudub materjali vaatamise õigus, suunatakse kasutaja otsingu lehele.

**Märkused:** Tavasituatsioonis ei tohiks kasutaja näha materjale, mille lugemisõigust tal pole. Kontroll vajalik URL aadressi käsitsi muutmise kontrollimiseks.

## **Kasutuslugu nr. 6 - Uue materjali lisamine**

**Olukorra kirjeldus:** Kasutaja lisab infosüsteemi uue materjali

**Eeltingimused:** Kasutaja on süsteemi sisse loginud vähemalt õpetaja õigustega

**Oodatav tulemus:** uus materjal on lisatud süsteemi ja ootab kinnitamist

**Kohustuslikud parameetrid:** õppematerjali nimi

**Valikulised parameetrid:** raskusaste, kasutatavad andurid, märksõnad, ülesande kirjeldus, ülesande/materjali failid, lahenduse kirjeldus, lahenduse failid, lahenduse peitmise valik, kuupäev, milleni lahendus peita, kasutajate nimekiri, kellele materjal näidatakse



**Põhivoog:**

1. Kasutaja valib menüüst “Lae materjal üles”.
2. Kasutaja valib, kas koostatakse ülesanne või õppematerjal.
  - a. Ülesanne
    - i. Kasutaja lisab üldandmed - nimi, raskusaste, kasutatavad andurid, märksõnad
    - ii. Kasutaja lisab ülesande kirjelduse ja soovi korral ülesande failid (kasutuslugu 6.1).
    - iii. Kasutaja lisab lahenduse ja soovi korral lahenduse failid (kasutuslugu 6.1)
    - iv. Kasutaja valib, kas materjali lahendus peita või mitte (kasutuslugu 6.2)
    - v. Kasutaja valib, kas materjali nähtavust piirata või mitte (kasutuslugu 6.3)
    - vi. Kasutaja vajutab nupule “Salvesta”.
  - b. Õppematerjal
    - i. Kasutaja lisab üldandmed - nimi, kasutatavad andurid, märksõnad
    - ii. Kasutaja lisab kommentaari
    - iii. Kasutaja lisab materjali failid
    - iv. Kasutaja vajutab nupule “Salvesta”
3. Süsteem salvestab andmed.
4. Kasutaja suunatakse “Minu materjalide” lehele.

**Alternatiivvoog: -**

**Tulemus veasituatsiooni korral:** Kuvatakse vastav veateade.

**Märkused:** Vaikeolekus lahendust ei peideta ja materjal on nähtav kõigile. Ülesande puhul on raskusastmeks “algaja”

**Kasutuslugu nr. 6.1 - Failide lisamine**

**Olukorra kirjeldus:** Kasutaja koostab materjali ja soovib lisada ülesande või lahenduse faili.

**Eeltingimused:** Kasutaja on süsteemi sisse loginud vähemalt õpetaja õigustega

**Oodatav tulemus:** Andmefailid on seotud loodava materjaliga

**Kohustuslikud parameetrid: -**

**Valikulised parameetrid: -**

**Põhivoog:**

1. Kasutaja vajutab üleslaadimise nupule (sõltuvalt brauseris “Browse”, “Choose”, “Choose file” jne) ja valib faili, mille ta tahab üles laadida.
2. Fail ilmub vastavalt “Ülesande failid” või “Lahenduse failid” kasti.
3. Kasutaja kordab tegevust, kui soovib üles laadida rohkem kui üks fail.

**Alternatiivvoog:**

**Tulemus veasituatsiooni korral:** Mingeid muudatusi ei tehta.

**Märkused:** Failid laetakse infosüsteemi peale kogu materjalivormi serverile saatmist.

**Kasutuslugu nr. 6.2 - Lahenduse peitmine**

**Olukorra kirjeldus:** Kasutaja koostab materjali ja soovib peita ülesande lahenduse

**Eeltingimused:** Kasutaja on süsteemi sisse loginud vähemalt õpetaja õigustega

**Oodatav tulemus:** Õpilased näevad lahendust alles peale valitud kuupäeva

**Kohustuslikud parameetrid:** Lahenduse avalikustamise kuupäev

**Valikulised parameetrid:** -

**Põhivoog:**

1. Kasutaja paneb linnukese “Peida lahendus” ette.
2. Kasutaja vajutab kuupäeva tekstiväljale.
3. Kasutajale kuvatakse raam kuupäeva ja kellaaja valikuga.
4. Kasutaja valib sobiva avalikustamise kuupäeva
5. Kasutaja vajutab “Sulge” nupule.

**Alternatiivvoog:**

**Tulemus veasituatsiooni korral:** Kuupäeva valiku raam kaob ja kasutaja peab kordama tegevust alates punktist 2.

**Märkused:** -

**Kasutuslugu nr. 6.3 - Nähtavuse muutmine**

**Olukorra kirjeldus:** Kasutaja koostab materjali ja soovib piirata materjali nähtavust

**Eeltingimused:** Kasutaja on süsteemi sisse loginud vähemalt õpetaja õigustega

**Oodatav tulemus:** Materjal on nähtav ainult valitud kasutajatele.

**Kohustuslikud parameetrid:** Kasutajad või grupid, kellele anda õigust materjali näha

**Valikulised parameetrid:** -

**Põhivoog:**

1. Kasutaja eemaldab linnukese “Materjal on avalikult kõigile nähtav.” eest.
2. Avanevad kaks tulpa - esimene sisaldab kõiki materjali loojaga seotud kasutajaid ja gruppe, teine on esialgu tühi.
3. Kasutaja valib õpilase nime või grupi, kellele ta tahab anda lugemisõiguse ja vajutab “>” nuppu.
4. Tegevust 3. korratakse, kuni 2. tulbas on kõik kasutajad ja grupid, kellele soovitakse anda õigus näha materjali.

**Alternatiivvoog:**

3. Et anda lugemisõigus kõigile õpilastele ja gruppidele, tuleb vajutada “>>” nupule.

**Tulemus veasituatsiooni korral:** Mingeid muutusi ei tehta ja kasutaja suunatakse punkti 3.

**Märkused:** Gruppide loomine ja kasutajate gruppidesse panemine toimub “Õpilased” lehel.

## **Kasutuslugu nr. 7 - Õpilaste lisamine**

**Olukorra kirjeldus:** Kasutaja soovib enda kontoga siduda teise kasutaja konto

**Eeltingimused:** Kasutaja on süsteemi sisse loginud vähemalt õpetaja õigustega

**Oodatav tulemus:** Valitud kasutajakonto on seotud ja nähtav “Õpilased” lehel kasutajate nimekirjas.

**Kohustuslikud parameetrid:** -

**Valikulised parameetrid:** Eesnimi, perenimi, kasutajanimi

**Põhivoog:**

1. Kasutaja valib menüüst “Õpilased” lehe
2. Kasutaja sisestab otsingusse märksõna
3. Süsteem otsib kasutajate andmebaasist kasutajaid, kelle eesnimi, perenimi või kasutajanimi sobivad märksõnaga.
4. Kasutajale kuvatakse otsingutulemus
5. Kasutaja valib vastava kasutaja juures lingi “Lisa”

**Alternatiivvoog:** -

**Tulemus veasituatsiooni korral:** Kuvatakse tühi tulemustetabel

**Märkused:** -

### **Kasutuslugu nr. 8 - Õpilaste ja gruppide lisamine gruppidesse**

**Olukorra kirjeldus:** Kasutaja lisab kasutaja või teise grupi olemasolevasse gruppi.

**Eeltingimused:** Kasutaja on süsteemi sisse loginud vähemalt õpetaja õigustega. Grupp, kuhu soovitakse kasutaja või grupp lisada on olemas.

**Oodatav tulemus:** Kasutaja või grupp on edukalt lisatud gruppi.

**Kohustuslikud parameetrid:** -

**Valikulised parameetrid:** -

**Põhivoog:**

1. Kasutaja valib menüüst "Õpilased" lehe
2. Kasutaja valib gruppide tabelist grupi, kuhu tahetakse uus kasutaja või grupp lisada ja vajutab "Lisa kasutajaid" lingile.
3. Avaneb uus aken, kus on tulbas kõik olemasolevad kasutajad ja grupid.
4. Kasutaja valib kasutajad ju grupid, keda ta tahab lisada, vajutades nende nimedele
5. Märgitud ridade tagataust muutub kollaseks.
6. Kasutaja vajutab nupule "Salvesta"
7. Aken sulgub ja "Õpilased" leht uueneb

**Alternatiivvoog:**

2. Kasutaja valib seotud kasutajate tabelist sobiva kasutajakonto ja vajutab "Lisa gruppe" lingile.
3. Avaneb uus aken, kus on tulbas kõik olemasolevad grupid.
4. Kasutaja valib grupid, kuhu kasutaja lisada.
5. Valitud ridade tagataust muutub kollaseks
6. Kasutaja vajutab nupule "Salvesta"
7. Aken sulgub ja "Õpilased" leht uueneb

**Tulemus veasituatsiooni korral:** Andmeid ei uuendata ja kasutaja peab kordama tegevust alates punktist 2.

**Märkused:** -

### **Kasutuslugu nr. 9 - Dokumentide kinnitamine avalikustamiseks**

**Olukorra kirjeldus:** Kasutaja kinnitab dokumendi korrektsuse, et selle olek muutuks avalikuks

**Eeltingimused:** Kasutaja on süsteemi sisse loginud vähemalt moderaatori õigustega. Dokument on loodud ja salvestatud.

**Oodatav tulemus:** Dokument on avalikustatud ja vastavate õigustega kasutajad saavad seda näha.

**Kohustuslikud parameetrid:** -

**Valikulised parameetrid:** -

**Põhivoog:**

1. Kasutaja valib menüüst “Kinnita dokumendid” lehe
2. Kasutaja valib tabelist soovitud dokumendi
3. Kasutaja tutvub dokumendiga ja avab saki (ingl. *tab*) “Kinnitamine”
4. Kasutaja veendub, et märgitud on “Kinnita dokument” otsus
5. Kasutaja vajutab “Salvesta dokumendi kinnitamise otsus” nupule
6. Kasutaja suunatakse kinnitamist vajavate dokumentide tabeli juurde (punkti 2.)

**Alternatiivvoog:**

4. Dokument ei vasta normidele.
5. Kasutaja valib “Dokument ei vasta normidele” valiku.
6. Kasutaja lisab tekkinud kommentaari tekstivälja selgituse
7. Kasutaja vajutab “Salvesta dokumendi kinnitamise otsus” nupule
8. Kasutaja suunatakse kinnitamist vajavate dokumentide tabeli juurde (punkti 2.)

**Tulemus veasituatsiooni korral:** -

**Märkused:** Kui dokument märgitakse mitte normidele vastavaks, siis vastav teade tekib alati dokumendi vaatamisel, kui kasutajaks on materjali looja või moderaator/administraator. Teade kaob, kui dokumenti teha muudatus.

## **Kasutuslugu nr. 10 - Kasutaja andmete muutmine**

**Olukorra kirjeldus:** Administraator soovib muuta mõne kasutaja andmeid

**Eeltingimused:** Administraator on süsteemi sisse loginud administraatori õigustega.

**Oodatav tulemus:** Kasutaja andmed on edukalt muudetud

**Kohustuslikud parameetrid:** -

**Valikulised parameetrid:** -

**Põhivoog:**

1. Administraator valib menüüst "Administraator" lehe
2. Administraator valib tabelist vastava kasutaja ja vajutab sellel real "Muuda" linki
3. Tekivad väljad kasutaja andmetega
4. Administraator teeb soovitud muudatused
5. Administraator vajutab "Ok" nupule.

**Alternatiivvoog:**

2. Kui kasutajaid on liiga palju, siis saab kasutada otsingut
  - a. Administraator sisestab otsingusse märksõna.
  - b. Kasutajad sorteeritakse vastavalt märksõnale - otsitakse eesnime, perenime, kasutajanime ja email tulpadest.
  - c. Jätkub punkt 2. tegevus

**Tulemus veasituatsiooni korral:** -

**Märkused:** -

## Lisa 3. Infosüsteemi lähtekood

Selle lõputööga on CD plaadil kaasas fail süsteem.zip, mis sisaldab loodud süsteemi RAJU keskusele.

Peale lahtipakkimist on süsteem järgneva struktuuriga:

- ckeditor - tekstiredaktori pistikprogrammi failid
- css - kujunduse failid
- files - süsteemis olevate materjalide failid
  - captcha - registreerimisel kontrollpildid
  - images - ülesannete, lahenduste või kommentaaride tekstiväljades lisatud pildid
  - materials - õppematerjalide failid
  - solutions - lahendustele kaasa pandud failid
  - tasks - ülesannetele kaasa pandud failid
  - tmp - materjalide loomisel failide ajutine asukoht
- fonts - tekstistiili failid (kasutusel registreerimise kontrollpildi loomisel)
- img - süsteemi disaini pildid
- js - JavaScript failid
- lib - JavaScript pistikprogrammi failid
- system
  - application
    - config - raamistiku häälestuse failid
    - controllers - süsteemi kontrollite failid
    - helpers - kontrollite poolt kasutatavate lisafunktsioonide failid
    - models - süsteemi mudelite failid
    - views - süsteemi vaadete failid
    - ... - muud raamistiku failid
  - ... - muud raamistiku failid
- uploadify - failide Flash üleslaadimise pistikprogrammi failid
- index.php

- database.sql - andmebaasi tabelite loomise fail
- license.txt - CodeIgniter raamistiku litsents