

Tallinna Ülikool

Matemaatika-loodusteaduskond

Informaatika osakond

Virgo Vardja

Veebipõhine helpdesk-rakendus

Bakalaureusetöö

Juhendaja: Jaagup Kippar

Autor: “” 2006.a.
Juhendaja: “” 2006.a.
Osakonnajuhataja: “” 2006.a.

Sisukord

Sissejuhatus.....	3
1. Alternatiivsed tarkvararakendused.....	5
1.1. Talisma Email.....	5
1.2. PHP Helpdesk.....	6
1.3. Hesk.....	6
1.4. phpBugTracker.....	6
2. Rakendus Helldesk.....	8
2.1 Nõuded rakendusele.....	8
2.2. Probleemi elutsükel – diagramm.....	10
2.3. Probleemi elutsükel – rakenduse sisene andmevoog.....	11
2.4. Kasutajaliides.....	12
2.4.1. Avatud ja lahendamisel probleemide nimekiri.....	12
2.4.2. Probleemi lisamine või redigeerimine.....	13
2.4.3. Kasutajate administreerimise liides.....	14
2.5. Andmebaasi struktuur.....	15
2.6. E-posti automaatne lisamine andmebaasi.....	17
Kokkuvõte.....	20
Summary: Web-Based IT Helpdesk Application.....	21
Viited.....	22
Kasutatud tarkvaraplatvorm.....	22
Võrreldud tarkvaralahendused.....	22
Rakenduse demonratsiooniversioon.....	22

Sissejuhatus

Helpdesk (ka *help desk*, kasutajatugi) on kahtlemata ülimalt oluline osa IT osakonna töös igas asutuses, kus IT-töötajaid on rohkem kui üks ning kasutajaid ja nende probleeme võib tekkida rohkem ning kiiremini kui IT osakond on suuteline jooksvalt lahendada. IT osakonna töötajatel peab olema lihtne saada ülevaadet asutuses tekkinud IT-alastest probleemidest. Samuti muudab kogu asutuse tööd lihtsamaks võimalus tekkinud probleeme sisestada infosüsteemi ning olla kindel, et probleem ei jää IT-töötajatel kahe silma vahele, nagu võib juhtuda selle e-posti teel edastamise puhul või, veelgi enam, IT-töötajate suusõnalisel teavitamisel.

Erinevaid *helpdesk*-rakendusi on loodud suurel hulgal ning paljude erinevate suurustega osakondade tarbeks. Samas on selge, et suurt, tuhandeid dollareid maksvat täielikku ja võimsat infosüsteemi ei ole mõtet juurutada väikeses asutuses, mille IT osakond võib koosneda vaid kahest-kolmest inimesest. Samuti ei piisa lihtsamast probleemide haldamise süsteemist suurtes asutustes, eriti kui selle asutuse tegevusala oluline osa ongi klientide probleemide lahendamine. Eesti mastaapides näiteks ei ole võimalik võrrelda väikeettevõtet (üks kontor ning paarkümmend töötajat) ja telekommunikatsioonikontserni (tuhandeid töötajaid ning sadu tuhandeid kliente). Samas on Eesti kohta suur ettevõtte tühiselt väike võrreldes maailma mastaapidega. Seetõttu tuleb iga ettevõtte *helpdesk*-lahenduse loomisel hinnata ettevõtte suurust ning vajadusi ja valida välja parajasti sobiv rakendus, mida kasutusele võtta.

Võib ka juhtuda, et juba olemasolevate rakenduste analüüsil selgub, et ükski pakutavatest ei sobi perfektselt konkreetse juhtumi puhul. Sel juhul võib osutada vajalikuks kas mõnd olemasolevat rakendust kohandada või luua hoopis uus lahendus. Selline olukord tekkiski järgnevalt kirjeldatava *helpdesk*-rakenduse loomise eel. Otsides näiteks otsingumootoriga Interneti avarustest *helpdesk*-tarkvara, pakutakse potentsiaalsele ostjale hulgaliselt erinevaid serverirakendusi, suurem osa neist tasulised. Teisest küljest leitavate vabavaraliste lahenduste plussiks on küll rahaliste kulude puudumine kuid miinuseks juurutamiseprobleemide tekke võimalus. Sel juhul on vajalik tõsiselt kaaluda, kas mitte luua täiesti uut tarkvararakendust, mis sobiks täpselt asutuse vajadustega ning mille looja saaks kasutajatele kiiresti ja hõlpsalt pakkuda tuge ning modifitseerida tarkvara, kui see nõutavaks osutub.

Kõnealusel juhul ja kõnealuses asutuses, kus ma sel hetkel *webmaster*'i ja süsteemadministraatori töökohustusi täitsin, jõudsin pärast olukorra kaalumist otsusele luua uus lihtne *helpdesk*-rakendus.

Käesolev bakalaureusetöö kirjeldab loodud tarkvara (koodnimega "*Helldesk*") planeerimist ning juurutamist, peatub põgusalt mõningate alternatiivide kirjeldustel, tuues välja põhjused, miks need alternatiivid ülesande täitmiseks ei sobi, ning tutvustab loodud rakendust. Kirjeldatud on pigem rakenduse käitumist ja kasutamist kui teda moodustavat koodi.

Rakendusest eksisteerib demonstratsiooniversioon, mille veebiaadress ning kasutaja-info asub bakalaureusetöö lõpus viidete jaotuses.

Hell Desk

Common mispronunciation of 'help desk', especially among people who have to answer phones at one.

(The Jargon File, version 4.4.7, <http://www.catb.org/jargon/>)

1. Alternatiivsed tarkvararakendused

Allpool on esitatud ülevaated mõnedest *helpdesk*-funktsionaalsust pakkuvatest rakendustest, millega olen ise kokku puutunud, neid administreerinud ja/või kasutanud. Kiire otsing leiab Internetis kümneid kui mitte sadu konkureerivaid tooteid, kuid tooteid tutvustav ja reklaamiv info ei pruugi olla usaldusväärne kaugeleulatuvate järelduste tegemiseks. Seega on taolised programmid vaatluse alt välja jäetud. Mainitud on mõnda vabavaralist lahendust, üht kommertstarkvaralist rakendust, mille haldamisvajadust pole õnneks tekkinud, kuid mille kasutamist ning häid ja halbu omadusi on demonstreerinud kolleegid. Samuti on kirjeldatud võimalikku stsenaariumi, mis kirjeldab, kuidas on võimalik juurutada ka tarkvara, mis pole originaalselt *helpdesk*-kasutuseks mõeldud.

1.1. Talisma Email

Talisma Email sobib suurepäraselt olukordadeks, kus firmal on kümneid tuhandeid kliente, kes päevast päeva probleemidega helpdeski poole pöörduvad. Üldjuhul on taolistel puhkudel ökonoomsem lahendus kasutada hulka eelnevalt määratletud vastusevorme, mis hõlmavad kõiki problemaatilisi alasid ning lahendusi erinevatele probleemidele. Talisma Email, mille kasutajaliides meenutab populaarset e-posti programmi Microsoft Outlook, võimaldab suurel kasutajatoomekonnal saada kiiret ülevaadet lahendamata probleemidest, klientidest, kes need probleemid on püstitanud, ja võimalikest kiirlahendustest neile probleemidele. Klienditoel on võimalus jälgida konkreetse kliendi probleemide ajalugu ning neile pakutud lahendusi. Juhtivtöötaja saab omakorda jälgida üksikute töötajate töökiirust ja -efektiivsust ning erinevate probleemide lahendamise kulgu. Iga konkreetse probleemi korral on näha ajavahemik, mil üks või teine töötaja selle lahendamisega tegeles.

Mõningatel juhtudel ei ole kliendihalduse vaatenurgast otstarbekas probleemide edastamine e-posti teel. Sel juhul võib näiteks luua veebivormi, milles klient lisaks enda probleemi kirjeldamisele võib selle ka juba eelnevalt kategoriseerida, misjärel automaatselt sisestatud probleemid süsteemis koheselt sorteeritakse ning erineva spetsialiseerumisega klienditeenindajad saavad koheselt lahendamisele võtta just nende tööalasse puutuvad probleemid.

Talisma on ülesehitatud klient-server rakendusena, kusjuures kliendiks ei sobi lihtne veebibrauser vaid klienditeenindaja arvutisse on tarvis paigaldada spetsiaalprogramm. Selline arhitektuur toob välja Talisma suurima probleemi – klient peab pidevalt olema serveriga ühenduses.

Kui ühendus katkeb, ei ole võimalik kasutada näiteks lokaalselt hoitavaid andmeid vaid klient-programm katkestab koheselt töö, parajasti pooleli olnud probleemi lahendus kaob ning ühenduse taastumisel tuleb seda uuesti lahendada hakata.

Talisma Email-rakenduse hind on töökoha kohta väljaostmisel 2500\$, või 350\$ kuutasu. See hind üksinda on piisav, et tema rakendamine kõnealusel juhul muuta mõttetuks.

1.2. PHP Helpdesk

PHP Helpdesk oli üks neist eksisteerivatest rakendustest, mis rohkem või vähem vastas otsitavatele vajadustele. Tegu on vabavaraplatvormil (PHP ja MySQL) toimiva lahendusega, mis on projekteeritud ning järk-järgult üles ehitatud asjaarmastaja poolt. Vastavalt autori vajaduste muutumisega on ka PHP Helpdesk'i võimalused suurenenud. Tarkvara analüüsil osutus, et selle juurutamiseks olemasolevasse keskkonda oleks tarvis läbi viia rohkem muudatusi kui ökonoomselt mõistlik. Lisaks arvestades potentsiaalset vajadust *helpdesk*-rakenduse muutmiseks ning laiendamiseks vastavalt tavapärase töö käigus tekkivatele vajadustele, otsustasin PHP Helpdesk'i juurutamisest loobuda.

1.3. Hesk

Hesk'i näol on tegu teise vabavaralise lahendusega, mis väljanägemiselt on lihtsam PHP Helpdesk'ist, kuid sisaldab samas piisavalt palju kasutusvõimalusi, et olla *helpdesk*-tarkvara otsingul atraktiivseks kandidaadiks. Kahjuks leidsin Hesk'i alles käesoleva bakalaureusetöö koostamise käigus olemasolevate *helpdesk*-rakendustega tutvudes. Arvestades platvormi väikest mahtu ning võrdlemisi lihtsat ülesehitust oleks Hesk olnud sobiv tarkvararakendus modifitseerimiseks ning selle järgseks juurutamiseks.

1.4. phpBugTracker

Kuigi veajälgimissüsteem ei ole otseselt mõeldud kasutajatoe rakenduseks, osutub, et mõningatel juhtudel on teda võimalik sobival viisil modifitseerides kasutajatoeks juurutada. Näiteks võib tuua tarkvarafirma, kus BugTracker-tarkvara on juba sügavalt juurdunud, mille töötajad oskavad ja tahavad seda kasutada ning kus veajälgimistarkvara ümberprogrammeerimine ei valmista ületamatuid probleeme.

phpBugTracker'i efektiivseks tööks tuleks rakenduse siseselt konkreetsed probleemid jagada projektidesse ning alamkategoriatesse. Juba juurutatud kategooriatele ei ole siis enam raske lisada näiteks uut kategooriat "Sise-IT". phpBugTracker sisaldab efektiivseid vahendeid jälgimaks konkreetsete probleemide või kategooriate aktiivsust ning olekut, mis hõlbustab kahtlemata nii IT osakonna töötajate kui siseklientide tööd.

Küll aga peab ära märkima phpBugTracker'i kaht puudust, mis tema kasutamist tavakeskkonnas võivad häirida.

Iga probleem sisaldab suure hulga erinevaid omadusi ehk andmebaasivälju. Tarkvara vigade haldamisel on need väga olulised kuid standardse IT alase probleemi puhul ei pruugi omada mõtet näiteks väljad "tarkvara versioon", "operatsioonisüsteem", "millises tarkvaraversioonis on prognoositud vea lahendamine" jne. Tavakasutaja jaoks võib kuvatud probleemi liigne infoküllus muutuda häirivaks ning segadusttekitavaks. Liigsete väljade eemaldamine on aga õnneks konkreetsel juhul võrdlemisi lihtne, kuna tegu on avatud lähtekoodis rakendusega.

Suurem probleem on aga phpBugTracker'i lähenemine turvalisusele. Süsteemi kasutajad paigutatakse gruppidesse ning erinevatele gruppidele antakse ligipääs erinevatele projektidele. Kui aga mõnel konkreetsel grupil on ligipääs projektile, siis võib sellesse gruppi kuuluv kasutaja sellesse projekti kuuluvate probleemidega sooritada kõiki operatsioone. See tähendab, et suvaline kasutaja, kel on ligipääs IT projektile (s.o. praktiliselt kõik asutuse töötajad) võib probleemi sulgeda või sellele määratud lahendaja ümber määrata.

Ülaltoodud puuduste tõttu tuleb tunnistada, et suurt hulka "tavakasutajaid" hõlmavas organisatsioonis phpBugTracker'i põhise helpdesk-süsteemi rakendamine võib põhjustada rohkem probleeme kui kasu. Seega on mõttekam juurutada pigem otseselt helpdesk-süsteemina töötama mõeldud tarkvara.

2. Rakendus *Helldesk*

2.1 Nõuded rakendusele.

Enne otsust kasutada mitte olemasolevat lahendust vaid rakendus vastavalt vajadusele luua, sõnastati järgmised nõuded, millele *helpdesk*-rakendus peab vastama.

Rakendus peab olema **vabavara**. Ei ole mõtet kulutada suurt summat, et hankida tarkvara, mida kasutatakse vaid mõne inimese poolt.

Rakendus peab **ühilduma olemasolevate süsteemidega**, s.o. toimima platvormil Debian GNU/Linux, Apache, PHP, MySQL.

On tarvilik, et probleeme oleks võimalik sisestada käsitsi (IT osakonna töötaja poolt) ning automaatselt, kui klient saadab e-kirja asutuse IT osakonna üldaadressile.

Ei ole tarvilik, et rakendus hõlmaks täielikku terviklahendust koos probleemide analüüsi võimalustega, erinevate allüksuste tekitamise võimalust, jms. Kuna enamus probleemidest saabub eeldatavasti e-posti vahendusel, on automaatne kategoriseerimine praktiliselt võimatu.

On tarvilik, et rakenduse veebiliides oleks seotud Apache-serveri poolse kasutajatuvastamisega, kasutades *Basic Authentication* meetodit. Kuna kõik keskkonnad, kus rakendust kasutama hakatakse, hõlmavad juba iseenesest serveripoolset kasutajatuvastust, pole mõtet luua mõnd muud, näiteks sessioonipõhist tuvastust, kasutavat süsteemi. Kui tehakse otsus eksisteeriva tarkvara rakendamise kasuks, tuleks seda modifitseerida nii, et kasutajatuvastus oleks ühtne ülejäänud platvormidega. Rakenduse-sisesed kasutajakontod tuleb luua eraldi ning siduda serveri poolt pakutavate kasutajanimedega. Selleks peaks rakendus sisaldama sobivaid vahendeid. Peab olema võimalik määrata rakenduse kasutajateks vaid osa serveri poolt autoriseeritud kasutajatest, ülejäänud kasutajad peavad saama veateate ning ei tohi saada rakendust kasutada.

On soovitatav, et kasutajaliidest oleks võimalik esitada mitmes erinevas keeles, ning et keeli oleks hõlbus vahetada nii rakenduse administraatori kui kasutaja vaatenurgast.

On tarvilik, et kasutajaõigused jaguneksid vähemalt kahte kategooriasse: tavakasutaja, kes saab probleeme lisada, lahendamisele võtta ning lahendamiseks määrata, ning administraator (IT osakonna juhataja), kes saab vajadusel probleeme ühelt lahendajalt teisele lahendamiseks määrata ning probleeme süsteemist täielikult eemaldada (näiteks kui tegu on rämpspostiga).

Kui osutus, et olemasolevad vabavaralised lahendused on kas ebasobivad või nõuavad esitatud vajadustele kohandamiseks liigselt palju ressursse, aega ja jõupingutusi, ning et ökonoomsem on luua just konkreetse olukorra lahendamiseks parajasti sobiv lahendus, püstitati veel järgmised eesmärgid:

Kasutatavaks programmeerimiskeeleks on veebirakenduse osal PHP. Veebiserveriks peaks kasutama Apache serverit, kusjuures ei ole oluline, kas tegu on vanema (1.3) või uuema versiooniga (2.0 või 2.2). Rakenduse kasutatavaks andmebaasimootoriks on MySQL versiooninumbriga vähemalt 4.0. E-kirja analüüsiva rakenduse puhul on vajalik PERL 5 ning lisamoodulitena MIME-Base64, MIME-tools, Text-Iconv, DBI ning DBD-mysql (kõik need moodulid on vabalt alla laaditavad PERLi moodulite repositooriumist CPAN).

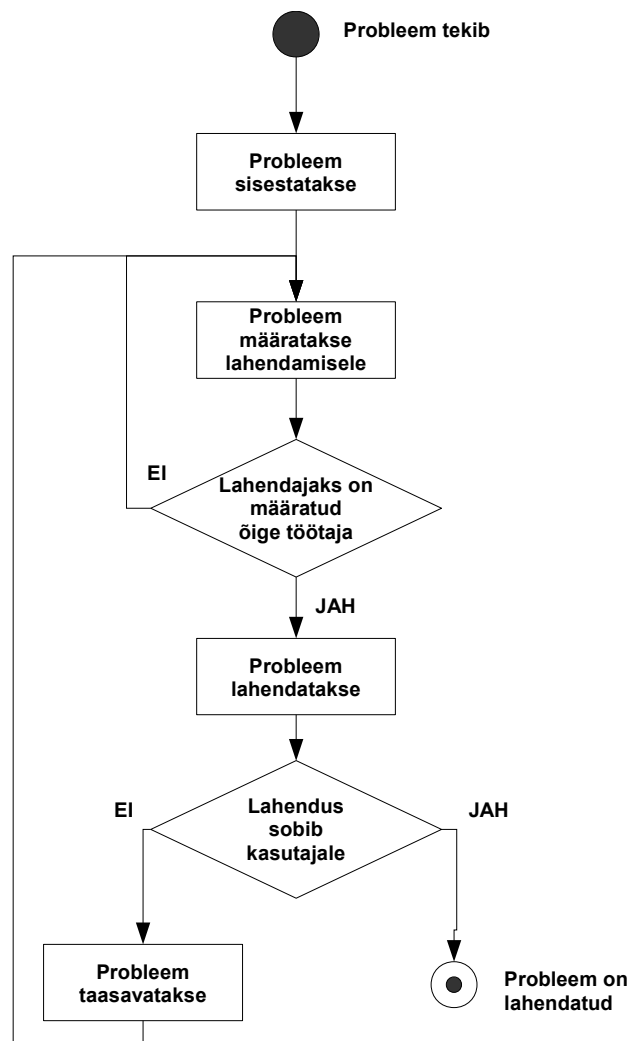
On soovitatav, et rakendus sisaldaks võimalikult vähe suuremahulisi kujunduselemente: visuaalne pilt peaks omama vähe graafikat ning olema üles ehitatud (D)HTML ja CSS vahenditega.

Eelmisest soovituselt tulenevana, **ei ole tarvilik**, et rakendus oleks ühtviisi kasutatav kõikvõimalikes keskkondades, kaasa arvatud vananenud brauseriversioonid (näiteks Netscape 4.7) ning mitte-graafilised brauserid (näiteks Lynx või W3M). Kasutatavateks brauseriteks on soovitatavalt Mozilla alates versioonist 1.5 ning derivatiivid (Firefox 1.0 või uuem); sobib ka Internet Explorer alates versioonist 5.5, Opera alates versioonist 7.0, Safari alates versioonist 1.2 jne.

On tarvilik, et rakendus oleks suuteline e-posti teel saabunud probleemidele saatma automaatse vastuse, kui kõnealune probleem esmakordselt lahendamisele võetakse.

Vastavalt ülaltoodud eeldustele loodi rakendus koodnimega *Helldesk*, mis käesoleva bakalaureusetöö kirjutamise ajal kannab versiooninumbrit 0.5.

2.2. Probleemi elutsükkel – diagramm



Joonis 1: Probleemi lahendamise protsessi olekudiagramm

Probleemi elutsükli kirjeldab joonis 1. Probleem sisestatakse kas IT osakonna töötaja poolt, saabunud e-kirja automaatsel töötlemisel, või mõnel muul võimalikul meetodil (näiteks asutuse intranetis paikneva veebivormi kaudu). IT osakonna juhataja kas määrab probleemi lahendamiseks mõnele töötajale, või töötaja võtab ise probleemi lahendamisele. Vajadusel määratakse probleem lahendamiseks mõnele teisele töötajale. Töötaja lahendab probleemi ning märgib selle lahendatuks. Kui avastatakse olukord, kus lahendatuks märgitud probleem on siiski aktuaalne, võib selle taasavada ning uuesti täitmisele määrata.

2.3. Probleemi elutsükkel – rakenduse sisene andmevoog

Probleemi lisamiseks andmebaasi on kaks võimalust.

1. Probleem saabub e-posti teel kindlale aadressile, misjärel ta edastatakse IT osakonna töötajatele ning lisatakse automaatselt andmebaasi lahendamata probleemina.
2. Probleem saabub telefoni teel, suusõnaliselt või muul meetodil, misjärel ta sisestatakse manuaalselt IT osakonna töötaja poolt andmebaasi.

Andmebaasis paiknev probleem on nähtav veebipõhise *helpdesk*-rakenduse pealehel "lahendamata probleemide" kategoorias. Probleem määratakse lahendamisele kas IT osakonna töötaja poolt (temale endale) või administraatori poolt (suvalisele rakenduse kasutajale). Siinkohal liigub probleem rakenduses "lahendamata probleemide" kategooriast "lahendamisel olevate probleemide" kategooriasse. Konkreetse probleemi lahendaja näeb seda lisaks ka "minu lahendamisel olevate probleemide" nimekirjas.

Esmase lahendamiselevõtmise korral võib vastavalt eelnevalt määratud tingimustele probleemi püstitajale saata automaatse e-kirja, milles IT osakond annab teada, millal probleem lahendamisele võeti ning kes lahendajaks on. Tingimuseks võib näiteks olla probleemi püstitaja e-posti aadressi kuulumine asutusesiseste aadresside hulka (näiteks kui asutuses on kasutusel aadressid kujul kasutaja@example.com, siis saadetakse automaatvastus probleemi püstitajale aadressiga johndoe@example.com kuid mitte isikule aadressiga janedoe@gmail.com).

Kui lahendaja on probleemi lahendanud, sulgeb ta probleemi, mis seejärel liigub süsteemis "lahendatud probleemide" kategooriasse vastaval rakenduse alamlehel. Sel puhul ei saadeta probleemi püstitajale automaatvastust vaid lahendaja teavitab ise vajalikke inimesi.



Kui osutub, et probleem püsib, võib nii tavakasutaja kui administraator selle "lahendatud probleemide" lehelt uuesti avada, misjärel see satub taas "lahendamisel probleemide" ning eelnevalt määratud lahendaja isiklikku probleemide nimekirja.

2.4. Kasutajaliides

2.4.1. Avatud ja lahendamisel probleemide nimekiri

Joonis 2 iseloomustab tüüpilist *helpdesk*-rakenduse peaakent nii, nagu see paistab administraatori-õigustes kasutajale.

Helpdesk

Lisa probleem						
Lahendamata probleemid						
ID	Prior	Informeerija	Aeg	Tekst		
2	Yellow	Virgo Vardja	23.04.2006 18:47	Suur probleem	X	
Minu probleemid						
ID	Prior	Informeerija	Aeg	Tekst		
3	Red	Kasutaja	23.04.2006 18:54	Server ei tööta	X	<input checked="" type="checkbox"/>
Lahendamisel probleemid						
ID	Prior	Informeerija	Aeg	Tekst	Lahendaja	
3	Red	Kasutaja	23.04.2006 18:54	Server ei tööta	Admin	X ✓
Lahendatud probleemid						
						Keel:  

[Kasutajate administreerimine](#)

Joonis 2: avatud ja lahendamisel probleemid

Kuvatud probleemide juures on hõlbus näha nende prioriteeti (rohelised toonid tähendavad madalaid, kollased keskmisi ja punased kõrgeid prioriteete), lahendajate nimesid ning seda, kas probleem on hetkel lahendamisel või mitte (roheline "linnuk" probleemi kõrval, mida probleemi lahendaja ise saab enda halduses olevatele probleemidele lisada või neilt eemaldada). Administraatoril on samuti õigus probleeme eemaldada (punane "X"). Nagu näha, on hõlpsasti ligipääsetavad ka probleemide lisamise, lahendatud probleemide, ning administraatoritele kasutajate halduse alamlehed. Klõps sinisel "ID" väljal lahendamata probleemide hulgas määrab klõpsaja koheselt ja hõlpsalt konkreetse probleemi lahendajaks. Liidese töökeele vahetamiseks tuleb sobivat keelt indikeerival ikoonil klõpsata. Keelte vahetamine toimib ainult juhul, kui kasutajal on

veebibrauseris lubatud küpsised (*cookies*), vastasel juhul on liides vaikimisi inglisekeelne. Kui kasutaja pole administraator, on ainus visuaalne erinevus kustutusnupu puudumine.

2.4.2. Probleemi lisamine või redigeerimine

Joonisel 3 on kujutatud probleemi lisamiseks või redigeerimiseks mõeldud liides, samuti administraatoriõigustes kasutajale.

Helpdesk: Probleem 3

Informeerija:	Kasutaja, <input type="text" value="kasutaja@example.com"/>	Prioriteet	<input checked="" type="radio"/> Kriitiline <input type="radio"/> Kõrge <input type="radio"/> Üle keskmise <input type="radio"/> Keskmise <input type="radio"/> Alla keskmise <input type="radio"/> Madal												
Probleem	Server ei tööta	Lahendaja	<input type="radio"/> Admin <input checked="" type="radio"/> User												
Tekst (E-maili sisu)	Suitsu tuleb	Staatuse:	Lahendamisele määratud												
Kommentaariid		<input type="button" value="Muuda"/> <input type="button" value="Määra ümber"/> <input type="button" value="Sulge"/>													
Ajalugu:	<input type="checkbox"/>	<table border="1"> <thead> <tr> <th>Aeg</th> <th>Kes</th> <th>Tegevus</th> <th>Kommentaariid</th> </tr> </thead> <tbody> <tr> <td>23.04.2006 18:54</td> <td>admin</td> <td>Lisatud</td> <td></td> </tr> <tr> <td>23.04.2006 19:19</td> <td>admin</td> <td>Ümber määratud</td> <td>to user</td> </tr> </tbody> </table>	Aeg	Kes	Tegevus	Kommentaariid	23.04.2006 18:54	admin	Lisatud		23.04.2006 19:19	admin	Ümber määratud	to user	
Aeg	Kes	Tegevus	Kommentaariid												
23.04.2006 18:54	admin	Lisatud													
23.04.2006 19:19	admin	Ümber määratud	to user												

Joonis 3: probleemi redigeerimine

Antud näites on probleem lisatud ning administraatori poolt töösse määratud kasutajale "User". "Lahendajate" nimekirjas kuvatakse kõiki süsteemi aktiivseid kasutajaid ning administraatoril on võimalus probleeme soovi korral uuele lahendajale määrata nupu "Määra ümber" abil. Kasutajate nimekirja ning "Määra ümber" nuppu ei kuvata tavakasutajale, nagu ka prioriteetide valikut. Konkreetset ekraanipildil on avatud probleemi ajalugu, kus on hõlbus näha probleemiga sooritatud operatsioone. Vajadusel võib peaaegu kõiki välja muuta: näiteks lisada kommentaare või

redigeerida automaatselt lisatud e-kirja sisu, mis võib sisaldada ebasoovitavaid osi (allkirjad, HTML-vormingus tekst vms).

2.4.3. Kasutajate administreerimise liides

Joonis 4 kirjeldab administraatorile mõeldud tööriista rakenduse kasutajate haldamiseks.

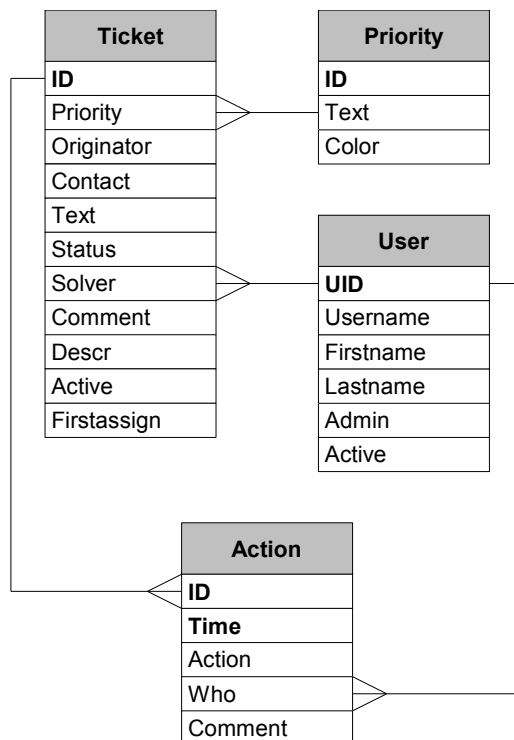
Helpdesk: Kasutajate administreerimine

UID	Kasutajanimi	Eesnimi	Perenimi	Adminn	Aktiivne
1	mail	Mail		Süsteemne kasutaja - ei saa muuta.	
2				Süsteemne kasutaja - ei saa muuta.	
3	<input type="text" value="admin"/>	<input type="text" value="Admin"/>	<input type="text" value="User"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	<input type="text" value="user"/>	<input type="text" value="User"/>	<input type="text" value="(Normal)"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
				<input type="button" value="Salvesta muudatused"/>	
Lisa uus kasutaja					
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
				<input type="button" value="Lisa"/>	

Joonis 4: kasutajate haldamine

Näha on süsteemselt defineeritud abikontod "mail" ja "" (tühi kasutajanimi). Neile lisaks on rakendusel üks administraator-kasutaja ning üks tavakasutaja. Administraatoril on võimalus olemasolevate kasutajate õigusi muuta ning uusi kasutajaid lisada. Kasutajate kustutamine ei ole võimalik vaid vajadusel tuleks kasutajad märkida mitteaktiivseks – selline teguviis ei lase lahendatud probleemide ajalool muutuda ebakorrektsesks.

2.5. Andmebaasi struktuur



Joonis 5: Andmebaasi olem-seos andmemudel

Rakenduse kasutatav andmebaas koosneb neljast andmetabelist (joonis 5).

Tsentraalseks tabeliks on *helpdeski* saabunud probleeme sisaldav tabel **Ticket**. Võrdlemisi staatilise abitabeli **Priority** andmete järgi defineeritakse konkreetse probleemi prioriteedi väli Priority (suhe tabelite Priority ja Ticket vahel: "üks mitmele"). Väljad Originator ning Contact sisaldavad probleemi püstitaja nime ja kontaktandmeid, kas käsitsi sisestatult või automaatselt e-kirjast eraldatult. Väljal Descr asub probleemi lühikirjeldus, milleks e-kirja puhul on teemarida. Väli Text võib sisaldada pikemat kirjeldust – e-kirja puhul kirja tekst ning väli Comment on mõeldud lahendajate poolsete kommentaaride salvestamiseks. Väli Status kirjeldab probleemi hetkeolekut (uus, lahendamisel, lahendatud jne), väli Active on indikaator, mis näitab, kas konkreetse probleemi lahendaja on probleemi märgistanud kui "hetkel lahendamisel" ning väli Firstassign on vajalik automatvastuse saatmiseks probleemi püstitajale.

Abitabel Priority kirjeldab probleemide võimalikke prioriteete, kusjuures selleks, et vajadusel oleks võimalik prioriteete lisada või eemaldada, ei kasutata prioriteedi märkimiseks tabelis Ticket

mitte prioriteeditabeli primaarset võtit (ID) vaid prioriteedi kirjelduse lühendit (väli Text). Näide kirjeldusest: prioriteet "Madal" ehk "Low" kannab identifikaatorit "LOW", prioriteet "Kriitiline" ehk "Critical" kannab identifikaatorit "CRIT" jne. Tabeli primaarvõtit kasutatakse üksnes prioriteetide sorteerimisel olulisuse järjekorras, mistõttu uute prioriteetide lisamisel tuleks tabel vajadusel ümber paigutada (näiteks uue prioriteedi lisamisel 1-CRIT ja 2-HIGH vahele tuleks alates prioriteedivõtmest 2 kõik prioriteetid ühe astme võrra allapoole nihutada, et tulemus oleks: 1-CRIT, 2-XXX, 3-HIGH jne).

Tabel User sisaldab infot kõikide rakenduse kasutajate kohta. Väljal Username on defineeritud kasutajanimed täpselt sellisel kujul, nagu nad on kasutusel serveripoolses kasutajatuvastuses. Väljad Firstname ning Lastname sisaldavad kasutaja ees- ja perenime (üldiselt kuvatakse üksnes eesnime, kuid kui tekib vajadus andmebaasi olekut kuvada klientidele, võib korrektsuse huvides olla vajalik näidata ka perenime). Väljad Admin ja Active defineerivad kasutaja oleku – kas tal on administreerimisõigus, ning kas temale on võimalik probleemi ümberregistreerida.

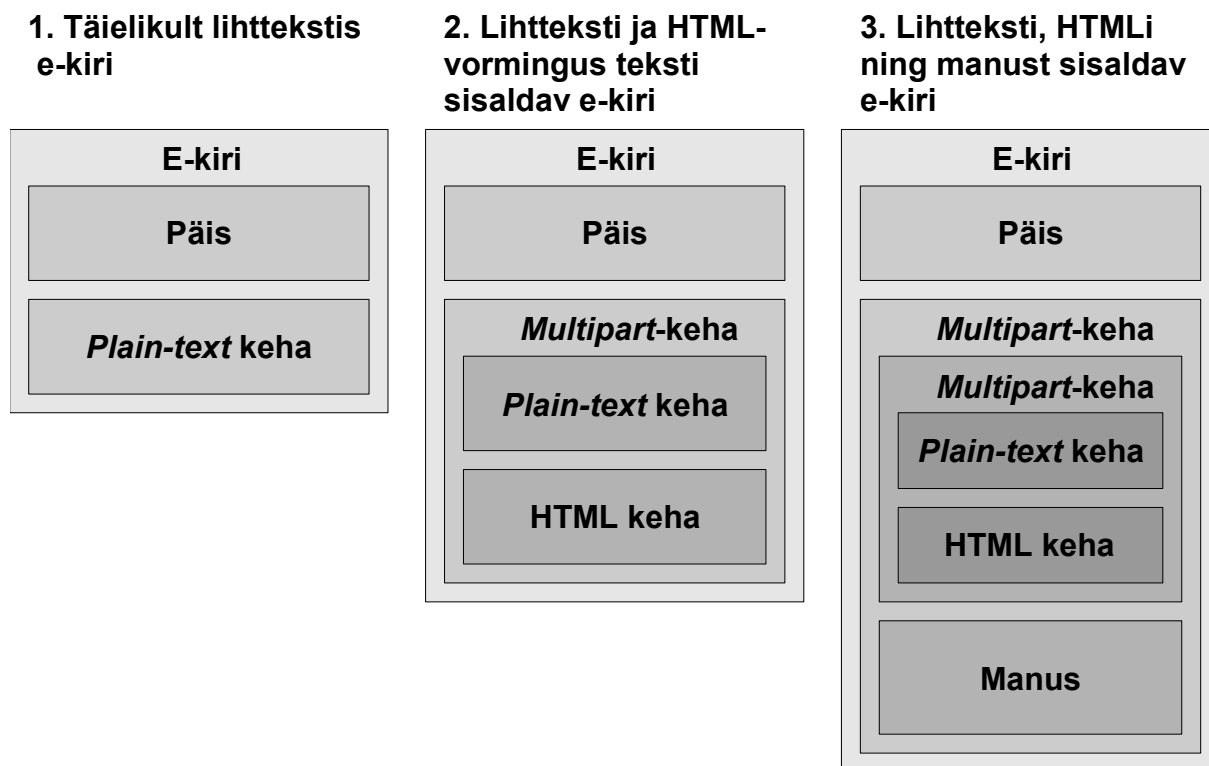
Kaks esimest kasutajakontot on süsteemselt reserveeritud. Kasutaja nr. 1 on e-posti automaatse lisamise tarkvara virtuaalkasutaja. Kasutaja nr. 2 on "tühi" kasutaja, mis on defineeritud rakenduse sisemiste protsesside lihtsustamiseks. Kumbki reserveeritud kasutaja pole administraatori- ega aktiivse omadusega.

Tabel **Action** kirjeldab kõiki probleemidega ette võetud operatsioone, alates lisamisest kuni probleemi sulgemiseni. Tegu on nõrga olemihulgaga, mille primaarvõti moodustub kahest esimesest väljast. Väli ID sisaldab probleemide primaarvõtmeid ning väli Time tegevuse toimumise kuupäeva ning kellaaega. Väli Action kirjeldab toimunud tegevust, väli Who sisaldab tegevuse sooritaja kasutaja-ID'd ning teatud tegevuste puhul (näiteks probleemi ümberregistreerimine) märgitakse väljale Comment lisainfo (kellele probleem ümber registreeriti).

2.6. E-posti automaatne lisamine andmebaasi.

Saabunud e-kirjade automaatne lisamine "lahendamata" probleemide hulka on võimalik PERL-skriptimiskeeles kirjutatud rakenduse abil. Lihtsaim meetod UNIX/Linux-keskkonnas kirjade suunamiseks skriptile töötlemiseks on kasutades tarkvara *procmal*.

Võimalikke laiemalt levinud e-kirja struktuure kirjeldab joonis 6. Mahuliselt väikseim on kahtlemata üksnes lihtteksti (*plain text*) sisaldav e-kiri, mida aga tänapäeval tänu kaasaegsete e-postiprogrammide võimalustele järjest vähem on võimalik kohata. HTML-vormingut sisaldavad e-kirjad on saamas pigem reeglilis kui erandiks. Nende struktuuri kirjeldab joonise 6 punkt 2 – nn. *multipart*-vormingus keha sisaldab korraga nii lihtteksti kui ka HTMLi. Kui e-kiri sisaldab veel ka manust (*attachment*), on vorming veelgi keerulisem (joonis 6 punkt 3): ühes *multipart*-osas sisaldub omakorda *multipart*-vormingus HTML- ja lihttekstis keha, ning ülejäänud *multipart*-osades e-kirjale lisatud manused.



Joonis 6: e-kirja vormingud

Kirja osade eraldamist demonstreerib järgnev koodilõik rakenduse e-posti töötleva osa failist **scanmail.pl**. Eelnevalt on muutuja `$text` seatud sisaldama kogu e-kirja sisu.

1. E-kiri jagatakse osadeks ning leitakse saatja aadress ning teema (*subject*).

```
$entity = $parser->parse_data($text);  
my $from = $from_addr = convert($entity->head->get('From'));  
my $subj = convert($entity->head->get('Subject'));
```

2. E-kirja keha jagatakse osadeks

```
my @parts = $entity->parts;
```

3. Leitakse keha lihttekst-osa. Vajadusel eraldatakse eelnevalt *multipart*-osa omakorda alamosadeks ning eraldatakse alles siis lihttekst-osa.

```
foreach $part (@parts) {  
    if ($part->head->mime_type =~ /^(multipart\/alternative)/i) {  
        @parts = $part->parts;  
    }  
}  
  
$notfound = 1;  
foreach $part (@parts) {  
    if ($notfound && $part->head->mime_type =~ /^(text|message)/i) {  
        my $io=$part->open("r");  
        while($io->read($buf,1024)) {  
            $body .= $buf;  
        }  
        $notfound = 0;  
    }  
}  
  
$body = $entity->stringify_body() if ($body eq "");
```

Ainult juhul, kui postiprogramm ei ole e-kirjale lisanud lihttekst-osa, tuleb salvestada kas kogu e-kiri või HTML osa ilma manuseta.

Hõlpsaim võimalus kirjade automaatse töötlemise seadistamiseks on tekitada rakenduse tööserverisse (www.example.com) süsteemne kasutajakonto, näiteks "helpdesk", mille e-posti aadressiks saab seega automaatselt "helpdesk@www.example.com". Toimiva procmail-tarkvara korral tuleb seejärel tekitada lihtne konfiguratsioonifail ".procmailrc":

```
:0  
| /home/helpdesk/scanmail.pl
```

See direktiiv suunab kõik saabuval e-kirjad automaatse töötlemise skripti sisendisse. Aadress "helpdesk@www.example.com" tuleks seejärel lisada IT osakonna üldadressile saadetavate kirjade adreesside hulka.

Nagu näha, on tänu GNU/Linux platvormile loodud vahendite kasutamisele, skriptimiskeele PERL laialdasele populaarsusele ning tema jaoks kirjutatud hulgalisi erinevaid ülesandeid täitvate moodulite saadavusele väga hõlbus e-kirju töödelda. Võrdluseks, üritades saavutada sama tulemust näiteks programmeerimiskeeles C või kasvõi rakenduse teistes osades kasutuses olnud PHP vahenditega, oleks programm muutunud mitmeid kordi pikemaks ning keerulisemaks.

Kokkuvõte

Rakendus *Helldesk* kasvas välja väikese IT osakonna vajadusest asutuses tekkinud probleeme jälgida ning neile operatiivselt vastata. Otsuse täiesti uue tarkvaralahenduse loomiseks võtsin koos ülejäänud osakonnaga vastu kui osutus, et ükski hetkel pakutavatest lahendustest meie vajadusi rahuldama ei sobi. Esimene versioon - 0.1 - alustas tööd 2005. aastal ning praegune versioon - 0.5 - on IT osakonna kasutuses tänapäevani.

Rakendus on mõeldud *helpdesk*'i põhivajaduste rahuldamiseks. Seetõttu sisaldab ta hõlpsasti haaratavat probleemide haldusliidest, kuid temas puuduvad näiteks vahendid probleemide sügavamaks kategoriseerimiseks ja analüüsiks. 2005. aasta neljandal veerandil tekkis küll kavatsus võimalusi laiendada ning rakendust integreerida planeeritud riistvara andmebaasiga, kuid nii minu enese kui teise planeeringusse kaasatud inimese lahkumise tõttu teisele ametikohale jäi see plaan realiseerimata. Kuna allesjäänud ning värskest tööle võetud töötajad ei ole avaldanud ka vajadust nende plaanide taastõstatamiseks ega esitanud muid täiendus- ja parandusettepanekuid, on *Helldesk*'i arendus hetkel efektiivselt külmutatud.

Viimaseks muudatuseks oli käesoleva bakalaureusetöö kirjutamise raames toimunud koodi korrastus ning e-posti automaatselt andmebaasi lisavas rakenduseosas leidunud vea (rekursiivse *multipart*-kirja töötlus) parandamine.

Kui tõepoolest tekib vajadus rakendust edasi arendada, näen võimalustena näiteks järgmiste potentsiaalsete funktsioonide loomist:

- Kategooriad, probleemide poolautomaatne kategoriseerimine saabunud e-kirja saatja ja/või teema järgi.
- Sidumine hüpoteetilise riistvara andmebaasiga, misjärel on võimalik teha analüüsi erinevate seadmete rikete sageduse kohta.
- Statistika probleemide lahendamise aja kohta ning ka probleemide haldajate ning nende ülesannete kategoriseerimise kohta.

Summary: Web-Based IT Helpdesk Application

The application lovingly titled "*Helldesk*" was born approximately a year ago, in 2005, out of the need for a web-based helpdesk application suitable for the tiny IT department of a small organization. After weighing the alternatives – buying expensive commercial software, finding and implementing a freeware alternative, and creating a brand new application, it was decided that the most effective solution would be number three – a brand new piece of software.

The company's unofficial standard was chosen for the platform that the new application was based on. This includes a server running Debian GNU/Linux operating system, Apache web server, PHP Hypertext Preprocessor and MySQL Database server for the Web-based front end, and Postfix e-mail server, Procmail mail processing software and PERL with a couple of additional modules for the automatic e-mail processing.

The newly-created software was created and implemented, and the IT department began to use it with great enthusiasm and success. In fact, after seeing the automatic mail handling and ticket assignment features, a version of the software was also implemented in another, unrelated department.

Helldesk's primary purpose is to collect and store e-mail messages that arrive to the company's internal IT support list. These messages are inserted into the application's database where they are assigned to the system's users. The user interface helps to clearly distinguish between open, assigned and closed tickets, thus allowing a single-glance overview of current cases.

While I am no longer personally working for the company in question, I have been informed that the IT department is still using *Helldesk* in a daily basis, while the other department has moved to a different platform (Microsoft Exchange and its public folder structure). The only two reasons I cannot implement this application in the company I currently work for, are the combination of a similar and therefore usable framework (BugTracker), and the intention not to introduce a different software platform.

Viited

Kasutatud tarkvaraplatvorm

- Debian GNU/Linux
<http://www.debian.org/>
- Apache HTTP server
<http://httpd.apache.org/>
- PHP Hypertext Preprocessor
<http://www.php.net/>
- MySQL
http://www.mysql.com/products/database/mysql/community_edition.html
- Postfix
<http://www.postfix.org/>
- Procmail
<http://www.procmail.org/>
- PERL ja CPAN
<http://www.perl.org/>
<http://cpan.perl.org/>

Võrreldud tarkvaralahendused

- Talisma Email
<http://www.talisma.com/>
- PHP Helpdesk
<http://phphelpdesk.sourceforge.net/>
- Hesk
<http://www.phpjunkyard.com/free-helpdesk-software.php>
- phpBugTracker
<http://phpbt.sourceforge.net/>

Rakenduse demonstratsiooniversioon

- <http://www.audentes.ee/~helldesk-demo/>
Administraatori õigustes kasutajanimi ja parool: admin / Admin
Tavakasutaja õigustes kasutajanimi ja parool: user / User
E-posti aadress: helldesk-demo@mail.audentes.ee