

Tallinna Ülikool
Informaatika Instituut

MS Silverlight arendusplatvormi uuenduste rakendamise juhend.

A Manual of Features of MS Silverlight Development Platform.

Bakalaureusetöö

Autor: Eiri Selberg
Juhendaja: Andrus Rinde

Autor: “ “ 2010. a.
Juhendaja: “ “ 2010. a.
Instituudi direktor: “ “ 2010. a.

Tallinn 2010

Autorideklaratsioon

Deklareerin, 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)

.....

(autor)

Sisukord

| | |
|---|----|
| Sissejuhatus | 4 |
| 1. Microsoft Silverlight tehnoloogia | 5 |
| 1.1. Silverlight 3 | 5 |
| 1.2. Silverlight 4 | 6 |
| 1.2.1. Silverlight 4 olulisemad uuendused | 6 |
| 2. Silverlight 4 tähtsamad uuendused | 8 |
| 2.1. Juurdepääs kohalikule kettale | 8 |
| 2.2. HTML sisu Silverlight´i sees | 9 |
| 2.3. Heli ja video salvestamine | 10 |
| 2.4. Veebikaamera ja mikrofoni signaali jagamine üle võrgu | 11 |
| 3. Uusi võimalusi tutvustavad näidisrakendused | 12 |
| Kokkuvõte | 14 |
| A Manual of Features of MS Silverlight Development Platform | 15 |
| Kasutatud kirjandus | 16 |
| Lühendite seletused | 18 |
| LISAD | 19 |
| Lisa 1. Uue Silverlight projekti loomine | 20 |
| Lisa 2. Kohalikult kettalt lugemine ja kirjutamine | 22 |
| Lisa 3. HTML sisu kuvamine Silverlight rakenduse sees | 26 |
| Lisa 4. Veebikaamera ja mikrofoni kasutamine | 29 |
| Lisa 5. Harjutusülesanded | 33 |
| a. Kohalikult kettalt lugemine ja kirjutamine | 33 |
| b. HTML sisu kuvamine Silverlight rakenduse sees | 34 |
| c. Veebikaamera ja mikrofoni kasutamine | 34 |

Sissejuhatus

Microsoft on jõudnud oma interaktiivsete veebilahenduste loomiseks mõeldud tehnoloogia Microsoft Silverlight (edaspidi Silverlight) 4-nda versioonini, mis kuulutati välja 15. Aprillil 2010. Uus versioon lubab kasutajatele mitmeid uusi võimalusi, mida varem oli võimalik saavutada vaid konkureerivat Adobe Flash tehnoloogiat kasutades ning ka selliseid, mida isegi Adobe Flash ei ole seni pakkunud. Tänu uutele võimalustele kogub tehnoloogia aina enam kuulsust ja kasutajaid, kes avastavad selle laiaulatuslike kasutusvõimalusi. Ka käesoleva töö autor avastas hiljuti enda jaoks selle tehnoloogia ja soovis sellega lähemalt tutvuda, mis andis tõuke antud töö kirjutamiseks.

Hoolimata kiirest arengust, mugavatest arendusvahenditest ja üsnagi laialdastest võimalustest, ei ole Silverlight veel nii palju kasutust leidnud kui võiks. Siiamaani kipub suurem osa veebiarendajaid interaktiivse, animeeritud sisu loomisel eelistama konkurenti Adobe Flash tehnoloogiat. Üheks peamiseks põhjuseks on teadmatus, kuna Silverlight tehnoloogia kasutusvõimalused ja kasutamisoskus ei ole jõudnud veel potentsiaalsete kasutajateni.

Töö eesmärgiks on veelgi enam tutvustada Silverlight tehnoloogiat, juhtides tähelepanu uutele võimalustele ning lihtsustada Eesti Silverlight autoritel üleminekut uuele versioonile just uute lisandunud võimaluste kasutamist silmas pidades. Töö kirjutamisel ja eesmärkide saavutamisel peab autor silmas eelkõige Microsoft Eesti ja BCS Koolituse koostöö raames läbiviidava Veebistuudiumi veebidisaini koolituse vajadusi ja samas ka Tallinna Ülikooli informaatika instituudi õppekavas olevaid kursuseid.

Eesmärkide saavutamiseks annab autor kirjanduse põhjal lühikese ülevaate Silverlight tehnoloogiast, ülevaate uuest versioonist 4, toob välja olulisemad uuendused ning annab selgitusi ja soovitusi mõningate olulisemaks peetud uuenduste kasutamise kohta. Silmas pidades Veebistuudiumi koolituse vajadusi, loob autor ka uuendusi käsitlevaid harjutusülesandeid.

1. Microsoft Silverlight tehnoloogia

Silverlight kuulutati esmakordselt välja 4. septembril 2007 aastal. Siiani on olnud tehnoloogia arendustsükli kestuseks üks aasta, mis tähendab, et igal aastal on oodata uut verisooni. Käesoleval ajal kasutatakse rohkem veel Silverlight 3 versiooni, kuna Silverlight 4 on täisversioonina väljas olnud vaid üürikest aega.

Silverlight tehnoloogia on interaktiivsete rikkasisuliste veebirakenduste (RIA) loomiseks, kasutades XAML keelt rakenduste kirjeldamiseks ning C# keelt programmiloojika lisamiseks.

1.1. Silverlight 3

Silverlight 3 beeta versiooni tutvustati avalikkusele 18. märtsil 2009 ning seda oli võimalik juba samal päeval Microsofti kodulehelt alla laadida. Lõplik versioon ilmus 9. juulil 2009.

Silverlight 3-es on võrreldes varasemate versioonidega suurenenud juhtelementide arv, juurde on tulnud järgnevad elemendid: andmete koordinaatvõrk (*DataGrid*), puuvaade (*TreeView*), andmevorm (*DataForm*) jne. Kolmandasse versiooni lisandusid ka navigatsiooni raamistikud (*Navigation Framework*), mis võimaldavad Silverlight rakendustel kasutada hüperlingitud navigatsiooni mudeleid ning sügavat sidumist (*deep linking*) teiste Silverlight rakendustega ehk siis on võimalik lisada järjehoidja (*Bookmark*) rikkasisulisele veebirakendusele (RIA). Meedia poolelt toetab Silverlight 3 uuendustena AAC (*Advanced Audio Coding*) audio dekodeerimist ja ka riistvaralise kiirendusega H.264 video dekodeerimist. Antud versioon pakub ka 1080p (HDTV) sujuvat voogesitust. Silverlight 3 toetab perspektiivset 3D-d, mis võimaldab 2D elemendid ümber kujundada 3D elementideks. Kasutusel on graafikaprotsessor (GPU), mis kiirendab visuaalse puu (*Visual Tree*) kompositsiooni. Silverlight 3 toetab ka *ClearType*'i teksti visualiseerimist (*render*).

Kolmanda versiooni kasutajaliidese (UI) elemendid toetavad elemendilt elemendile sidumist, mis võimaldab ühe elemendi siduda teise elemendi seisundiga, ja samuti andmete sidumise valideerimise mehhanismi. Erinevalt Silverlight 2-st, kus sai faile salvestada ainult isoleeritud andmehoidlasse, saab Silverlight 3 rakendusi salvestada suvalisse kohta failisüsteemis, faili salvestamise dialoogi (*Save File Dialog*) kaudu. Samas arvutis töötavate rakendustega suhtlemiseks ja võrguühenduste jälgimiseks kasutab Silverlight kohaliku ühenduse (*LocalConnection*) API-t. Silverlight 3 toetab väljaspool brauserit (*Out-of-Browser*) töötamist,

st Silverlight'i rakendusi on võimalik installeerida süsteemi nii, et rakendustele on ühenduseta juurdepääs ning rakendus käivitub väljaspool brauserit. (Microsoft Silverlight, 2009)

Kuna Microsoft arendab pidevalt antud tehnoloogiat edasi siis kehtivad eelnevad teadmised ka igas uuemas versioonis, mis tähendab seda, et uue versiooni kasutamine on lihtsam neile, kes on ka eelnevalt tegelema Silverlight rakendustega.

1.2. Silverlight 4

18. novembril 2009 professionaalsete arendajate konverentsil (PDC) Los Angeles'es avalikustas Microsoft Corporation Silverlight 4 beeta versiooni. 15. Märtsil 2010 ilmus RC versioon ning lõppversiooni ilmus käesoleva aasta 15. Aprillil.

Silverlight 4-s versioon pakub kasutajale palju erinevaid uusi funktsioone ning juhtelemente, mis võimaldavad arendajatel, kasutades erinevaid kombinatsioone erinevates Microsoft arendusvahendites (Microsoft Visual Studio 2010 ja Microsoft Expression Blend 4 beeta), kiirelt arendada huvitavaid ja võimsaid rakendusi (Microsoft Silverlight, 2009)

1.2.1. Silverlight 4 olulisemad uuendused

Järgnevalt on välja toodud 10 olulisemat uuendust, millest on palju räägitud Silverlight'i foorumis (<http://forums.silverlight.net>) ning mis olid kasutajate poolt enim oodatud. Lisaks toob autor 11-nda uuendusena välja endale silma jäänud uuenduse, mis tasuks märkimist. Mitmete tarkvaraarendajate nagu näiteks GoYello (<http://blog.goyello.com>) IT firma, kus tegeletakse erinevate veebirakenduste loomisega, meeskonna arvates on järgnevad uuendused kõige huvitavamad:

1. Juurdepääs veebikaamerale ja mikrofonile, mis võimaldab jagada video ja heli. Seda saab kasutada suhtlemiseks näiteks klienditeeninduse rakendustes. Enne tundsid kasutajad sellest puudust ja sellest tulenevalt oli Silverlight nõrgemal positsioonil, kui konkurendiks olev Adobe Flash.
2. Võimalus heli ja video kohalikule kettale salvestada. See uuendus võimaldab ilma serveriga suhtlemata luua kompressioonita videofaile (*RAW Video*), mis võimaldab lõppkasutajale laiaulatusliku suhtluse ja suhtlemise stsenaariume, näiteks videokonverentsid.

3. Juurdepääs kasutaja kohalikule kettale, mis võimaldab failide lugemise ja kirjutamise, kindlatest kettal olevatest kataloogidest. Kui eelnevalt töötasid rakendused nagu liivakastis (*Sandboxed*), millest väljapoole pöördumine oli turvalisuse eesmärgil blokeeritud, siis nüüd on võimalik luua liivakastist väljas olevaid rakendusi (*Out of Sandbox Applications*), millel on seejuures tagatud ka turvalisus.
4. Võimaldab andmeid välistest allikatest oma rakendusse üle tuua kasutades kas kopeeri ja kleebi võimalust (*Copy and Paste*) või pukseerimist (*Drag and Drop*)
5. Täienenud tugi hiiresündmustele, näiteks parema klõpsu kontekstimenüü (*Right-Click Context Menu*) ja hiire kerimisnupu kasutamine.
6. Multiedastuse võrgustik (*Multicast Networking*), mis võimaldab ettevõtetel vähendada kulutusi voogesituste levitamisel (*Streaming Broadcast*), nagu näiteks firma koosolekud ja koolitusprogrammid.
7. Võimalik käivitada teisi töölaual (*Desktop*) asetsevaid programme nagu Office, näiteks taotleda e-kirja saatmist Outlook´ilt, saata aruanne Word´i või andmed Excel tabelisse.
8. COM automatiseerimine, mis võimaldab rakendusel juurdepääsu seadmetele ja teistele süsteemi omadustele, kutsudes need välja rakenduse komponentidesse, näiteks USB turvakaardi lugeja (*Security Card Reader*).
9. Põhjalik printimise tugi, mis võimaldab aruandeid ja dokumente printida paberkandjale ning vaadata virtuaalset printimise vaadet (*Virtual Print View*) sõltumata ekraani sisust.
10. .NET keeleülene käitusajamoodul (CLR), mis võimaldab kompileeritud koodi käivitada nii töölauarakendusena kui ka brauseris oleva Silverlight´i rakendusena.
11. HTML elemente paigutada rakenduse sisse, mis võimaldab tugevama liitmise võime veebiserverite sisuga, näiteks e-kiri, abi ja aruanded.

(Greñ, 2009)

2. Silverlight 4 tähtsamad uuendused

Silverlight 4 toob kasutajani palju erinevaid ja huvitavaid uuendusi, kuid Eesti Silverlight asjatundjate hinnangul on järgnevad neli uuendust just need, mida võiks lähemalt uurida ja lugejale tutvustada.

2.1. Juurdepääs kohalikule kettale

Siiani ei olnud võimalik Silverlight rakendustega pääseda ligi kasutaja arvuti kohalikule kettale, kuna kõik rakendusi oli turvalisuse huvides võimalik käivitada oma kindlates piirides, mida nimetati „liivakastiks“ (*Sandbox*). Nüüd aga on rakendused viidud „liivakastist“ välja (*Out of Sandbox*) ning turvalisuse tagamiseks saab rakendusi kasutada vaid kõrgendatud õigustega (*Elevated Trust*) brauseri välistes (OOB) rakendustes. See tähendab, et rakendus käivitub kohalikus arvutis ilma interneti ühenduseta ning tänu sellele pääseb rakendus ligi kasutaja kohalikule kettale.

Silverlight versioonides 2 ja 3 olid juba mõned põhilised faili juurdepääsu funktsioonid. Alates versioonist 2 on võimalik kasutada faili avamise dialoogi (*OpenFileDialog*), mis annab kasutajale ainult lugemisõiguse (*read-only stream*) valitud failile, see tähendab, et ainus, mis kasutajal oli lubatud failiga teha, oli lugemine ja rakenduses selle sisu kasutamine. Versiooniga 3 lisandus faili salvestamise dialoog (*SaveFileDialog*), mis üllatuslikult lubab faile salvestada kohalikule kettale. Versiooniga 4 lisati juurdepääs andmetele, mis asuvad kas kataloogis Dokumendid (*My Documents*), Pildid (*My Pictures*), Videod (*My Videos*) või Muusika (*My Music*) (Mac´il samamoodi). Nendes kataloogides on võimalik lugeda, luua ja kustutada faile, kusjuures failidega töötamine ei ole erinev tavalises .NET-is failidega töötamisest. Oluline on see, et rakendus peab olema usaldusväärne, mis tähendab seda, et see peab olema käivitatud brauseri väliselt ning kõrgendatud õigustega. Silverlight´il on kindel loenduse funktsioon: *Environment.SpecialFolder*, mida saab kasutada faili füüsilise asukoha määramiseks. Kuigi selle funktsiooniga ei ole võimalik kõikidesse kataloogidesse pääseda leiab funktsioon üles erinevad kataloogid, mis paiknevad kasutaja arvutis. Sealhulgas leiab funktsioon ka näiteks kataloogi Programmifailid (*ProgramFiles*), kuid kui kasutaja üritab midagi antud kataloogis teha, ei ole see võimalik.

(Cleeren, 2009)

2.2. HTML sisu Silverlight'i sees

Seni on Silverlight'i ja HTML-i sidumisel olnud võimalik vaid Silverlight'i lisamine veebilehele ja seda kahel moel:

- Täisekraani ulatuses (*Screen-Filling*) Silverlight'i rakendus.
- Silverlight'i „saared“ (*Silverlight “Islands”*)- väike(sed) Silverlight'i manus(ed) veebilehel.

Silverlight rakendustes HTML sisu kasutamine ei olnud siiani võimalik, kuid Silverlight 4-ga lisandus uus võimalus Silverlight'i ja HTML-i kokku toomiseks, veebibrauseri juhtelemendi (*WebBrowser Control*) abil. See lubab kasutajal näha Silverlight'i rakenduse sees HTML-i sisu. Antud juhtelemendi käsitlemisel on kõige tähtsamad järgnevad atribuudid:

- *Source* (Allikas): saab või seab URI, mis peaksid olema veebibrauseri juhtelemendi visualiseeritud.
- *Navigate* (Navigeerimine): määrab URI, mille sisu peaks olema juhtelemendi laetud (töötab identselt allika atribuutidega (*Source Property*)).
- *NavigateToString* (Navigeeri stringiks): kuvab lennult genereeritud HTML-i stringi. Seda saab sooritada antud meetodit kasutades.

Veebibrauseri juhtelement töötab ainult siis, kui Silverlight'i rakendus on käivitatud brauseriväliselt (OOB). Kui püüda käivitada seda brauseri sees, kuvatakse veateade kirjaga: *HML is Disabled* (HTML on keelatud).

(Cleeren, 2009)

Veebibrauseri juhtelemendi juures on üks huvitavaim aspekte see, et on võimalik kasutada tugiskripti (*InvokeScript*) meetodit, mis lubab välja kutsuda JavaScript'i funktsioonide sisu, mis on juhtelemendi poolt hostitud. Seevastu JavaScript'is, mis on võõrustatud veebibrauseri juhtelemendis, saab kasutada akende välist teavitamist (*window.external.Notify*) selleks, et tõsta skripti teavitamissündmusi (*ScriptNotify events*).

(Prosise, 2009)

Lisaks veebibrauseri juhtelemendile pakub Silverlight 4 ka veebibrauseripintslit (*WebBrowserBrush*), beeta versioonis kandis antud funktsioon nime HTML-pintsel (*HTMLBrush*). See on uus pintslit klass, mis võimaldab katta ükskõik millise kasutajaliidese elemendi, mille sisu visualiseeritakse, veebibrauseri juhtelemendiga (Golesh, 2009).

Üheks erinevuseks kahe eelpool mainitud HTML-i kasutatava funktsiooni juures on see, et veebibrauseri juhtelement võimaldab kuvada reaalses muutuvat sisu, aga veebibrauseripintsel seda ei võimalda. Seega võib järeldada, et veebibrauseripintslit kasutatakse visuaalsete efektide loomiseks, mis sisaldavad HTML sisu, näiteks nagu HTML piltmõistatus (*puzzle*) (vt. Joonis 1), mida esitleti professionaalsete arendajate konverentsil (PDC). Ning selleks, et kuvada reaalses interaktiivset HTML sisu, on mõttekas kasutada veebibrauseri juhtelementi. (Prosize, 2009)



Joonis 1. HTML piltmõistatuse rakenduse algvorm ja piltmõistatuse vorm

2.3. Heli ja video salvestamine

Peale seda, kui ilmus esimene Silverlight'i versioon, on kasutajad kannatlikult oodanud sellist Silverlight'i versiooni, mis võimaldab kasutajale ligipääsu nii veebikaamerale kui ka mikrofonile. Konkurendiks olev Adobe Flash pakub selliseid võimalusi juba ammu.

Silverlight 4 aga kõrvaldab selle puuduse. Kasutajad peavad märkimisväärseks seda, et antud tehnoloogia võimaldab stsenaariume, mida konkurendiks olev Adobe Flash ei võimalda, näiteks on võimalik Silverlight'iga arendada rakendusi *Windows Phone 7 Series* jaoks. (Neubeck, 2009) Antud uuendus annab palju huvitavaid võimalusi, näiteks on võimalik luua rakendus, mis loeb veebikaamera ees hoitavate toodete vootkoode ja otsib tootekirjeldust (Cleeren, 2009).

Silverlight 4 võimaldab arendajatele juurdepääsu kasutaja arvutisse salvestatud kompressioonita heli- ja videofailidele rakendustega, mis töötavad nii brauseri sees kui ka väljas. Kui heli või video soovitakse rakenduse poolt esmakordselt käivitada, peab kasutaja seda lubama. Antud päring kindlustab, et rakendus ei pääse seadmetele ligi ilma kasutaja teadmata ning see tagab turvalisuse, kuna kasutaja teab täpselt, milline rakendus soovib veebikaamerat ja mikrofoni kasutada. Kui arvutil on mitu seadet, saab kasutaja valida

vaikimisi meelepärase veebikaamera ja mikrofoni, kasutades Silverlight'i plugina (*Plug-in*) veebikaamera/mikrofoni vahekaarti.

(Yack, 2009)

Videopintsel (*VideoBrush*) on üks põnevamaid Silverlight'i meedia funktsioone, mis oli ka varasemates versioonides, võimaldades katta meedia elemente vaid piiritletult, näiteks mängiti videovooge välisallikatest. Kuid uue versiooniga annab antud funktsioon kasutajale võimaluse katta ükskõik milline ala videopildiga. Silverlight 4 võimaldab kasutada videopintsliga ka veebikaamerast püütavat reaajas olevat videot. Seega oli varasemates Silverlight'i versioonides meedia elementide katmine videoga piiratud. Videopintslit kasutati enamasti selleks, et luua põnevaid ja huvitavaid efekte, näiteks video mängimise peegeldused.

(Neubeck, 2009)

2.4. Veebikaamera ja mikrofoni signaali jagamine üle võrgu

Microsoft on teinud Silverlight 4-ga esimese sammu selle poole, et oleks võimalik pidada videokonverentsi, mis tähendab seda, et üle interneti edastatakse reaajas mikrofoni heli ja veebikaamerast pilt. See on küllaltki suur edasimineku, kuid lahendamist vajavad veel järgnevad probleemid:

- H.263 ja H.264 kõrglahutusega (HD) audio ja video tihendamine ei ole kättesaadav.
- Puudub reaaja transpordiprotokolli tugi, nagu näiteks RTP seetõttu on kolm võimalust, kas kasutada transpordiprotokollina: TCP-d, UDP-d või siis teoreetiliselt on innukamatel arendajatel võimalus programmeerida ise transpordi protokoll.
- Puudub spetsiaalne voogesitusserver Silverlight'ist heli ja video otseseks esituseks meedia serverisse ja meedia serverist Silverlight'i klientidele. Puuduse kompenseerimiseks on kaks lahendust, kas kasutada Microsoft Windows Media Serverit või on teoreetiliselt võimalik innukamatel kasutajatel luua oma server.

(Abdelqader, 2010)

Hetkel tundub videokonverents Silverlight 4-ga veel kaugel tulevik, kuna Microsoftil ei ole plaaniski avalikustada allikaid, mis võimaldaksid video kodeerimist ja voogesitust. Seni ei ole ka keegi usinamatest arendajatest oma lahendusi avalikustanud.

3. Uusi võimalusi tutvustavad näidisrakendused

Üheks töö eesmärgiks on luua töös käsitletud uuenduste kohta harjutusülesanded, mis annaksid lugejale võimaluse luua huvitavaid Silverlight'i rakendusi. Järgnevates lõikudes tutvustab autor lugejale kuidas on võimalik uuendusi kasutada ning valmistab tutvutuse käigus näidisrakendused ja pakub välja harjutusülesanded (vt. Lisa 5), mille abil on lugejal võimalik proovida iseseisvalt uuenduste võimalusi.

Selleks, et luua Silverlight tehnoloogia 4-da versiooniga rakendusi on arendajal tarvis, kas Microsoft Expression Blend 4 RC või Microsoft Visual Studio 2010 ning Silverlight 4 tarkvara, mida on võimalik alla laadida järgnevalt veebilehelt: <http://www.silverlight.net/getstarted>. Samuti on tasuta saadaval ka *Deep Zoom Composer*, mis on Silverlight'i tehnoloogia interaktiivsete piltide ja pildialbumite loomiseks.

Autor loob veebivälises režiimis töötava näidisrakenduse, mille abil on võimalik kasutajal kohalikult kettalt faile lugeda ja kirjutada. Kuna eelnevalt töötasid rakendused ainult liivakastis, siis leidis autor, et antud rakendus tutvustab kõige paremini uut, liivakastist väljas oleva rakenduse loomise võimalust. Samuti tutvustab uut funktsiooni *Environment.SpecialFolder*, mille abil on võimalik juurdepääs järgnevatele kataloogidele ja nende sisule: Dokumendid, Muusika, Pildid ja Videod. (vt. 0)

Järgmiseks näidisrakenduseks on rakendus, kus kuvatakse HTML sisu rakenduse sees, antud rakendus on sarnaselt eelmisele näidisele veebiväline rakendus, kuna veebibrauseri juhtelementi (*WebBrowser Control*) ei ole võimalik kasutada brauseri siseselt ehk siis ei ole võimalik Silverlight'i rakenduses veebis kuvada HTML sisu, kuid brauseriväliselt on see võimalik. Samuti on rakenduses kasutatud ka uut veebibrauseri pintslit (*WebBrowserBrush*), mis võimaldab kuvada rakenduses veebilehte. Kasutajal on võimalik sisestada aadressiväljale uue lehe aadress ning rakendus kuvab soovitud lehe, millel on võimalik ringi vaadata, navigeerida. Autor soovis siduda antud rakenduse esimese näidisega, seega oli plaanis salvestada soovitud veebilehe sisu kasutaja kohalikule kettale, näiteks kokanduse teemaline rakendus, mis võimaldaks veebist retsepte hiljem kasutamiseks salvestada. Kuid rakenduse loomise käigus selgus, et salvestada on võimalik vaid seda lehte, mille täpne aadress on sisestatud aadressi väljale, kuna linke järgides navigeerimise järel ei ole võimalik pärida hetkel aktiivset URI-d. Kuigi Silverlight 4-a veebis olevas juhendis on olemas funktsioonid

Navigated ja *Refresh*, ei ole võimalik neid kasutada, kuna funktsiooni sisestamisel esineb kompileerimistõrge. (vt. Lisa 3)

Kolmandas näidiskrakenduses tutvustab autor veebikaamera ja mikrofoni kasutamist. Kuna Silverlight'i arendajad on selle uuenduse kohta andnud enamasti positiivset tagasisidet ja seda uuendust juba kaua oodatud, arvas ka autor, et näidiskrakenduse tegemine on vajalik. Rakenduses on kasutajal võimalus valida arvutiga ühendatud veebikaamera ja mikrofoni juhul, kui arvutiga on ühendatud rohkem seadmeid. Seejärel kuvab rakendus reaajas veebikaamerast pildi videopintsi (*VideoBrush*) abiga. Videot ja heli on võimalik salvestada, kuid selleks peab rakendus olema rakendus käivitatud brauserivälises režiimis. Hetkel on võimalik salvestada vaid kompressioonita heli- ja videofaile, kuigi usinamad arendajad on juba ka loonud lahendusi, mis võimaldavad salvestada pilte JPEG või PNG vormingus ja videoid AVI vormingus. Eelpool mainitud arendajad jagavad lahkelt oma lahendusi ka teistele ning neid on võimalik leida veebist ning oma projektidesse importida ning edukalt kasutada. (vt. Lisa 4)

Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli tutvustada lugejale Silverlight tehnoloogia uusi võimalusi ja luua nende kohta näidisrakendused, mille abil on Eesti Silverlight tehnoloogia huvilistel lihtsam üle minna uuele versioonile.

Esimeses peatükis tutvustab autor praeguse ajani Silverlight tehnoloogia kõige enam levinud versiooni 3, kuna iga uuema versiooni kasutamisel tulevad eelnevate versioonide kasutamise teadmised ainult kasuks. Autor toob välja uue, hiljuti avalikustatud 4-da versiooni 10 uuendust, mis on leidnud Silverlight'i foorumis enim kõneainet. Lisaks toob autor välja endale olulisena tundunud uuenduse.

Teises peatükis tutvustab autor Eesti Silverlight arendajate arvates nelja kõige huvitavamat uuendust, milleks on juurdepääs kohalikule kettale, HTML sisu kuvamine Silverlight rakenduses, heli ja video salvestamine ning veebikaamera ja mikrofoni signaali jagamine üle võrgu.

Kolmandas peatükis tutvustab autor uuenduste käsitlemise võimalusi ning loob iga eelnevalt väljatoodud uuenduse kohta näidisrakenduse, mille seletused asuvad antud töö lisades. Samuti pakub autor välja harjutusülesanded, mis võimaldavad õppuritel iseseisvalt proovida uuenduste võimalusi.

Käesoleva töö praktilises osast käsitletakse järgmiseid teemasid: uue Silverlight projekti loomine, kohalikult kettalt lugemine ja kirjutamine, HTML sisu kuvamine Silverlight rakenduse sees, veebikaamera ja mikrofoni kasutamine ning harjutusülesanded

Kahjuks esines töö tegemise käigus ka erinevaid probleeme, nimelt ei olnud võimalik ühilduvusprobleemide tõttu installeerida ühte masinasse nii Visual Studio 2010 kui Expression Blend 4 beetat, kuna seejärel need programmid enam ei käivitunud. Samuti on probleeme mõnede uute funktsioonide kasutamisega, näiteks veebilehe URI uuendamise funktsioonid. Tegemist on tootja vigadega ning need on lubatud peatselt lahendada.

Käesoleva bakalaureusetöö jätkuna tuleb autoril näidisrakenduste loomise kirjeldused kohandada, et need käsitleksid rakenduste loomist kasutades Expression Blend 4 versiooni, et koostatud näidised sobiksid veebistuudiumi koolituse tarbeks.

Kuna Silverlight on väga kiiresti arenev tehnoloogia ning igal aastal on oodata uut versiooni arvab autor, et hoiab antud tehnoloogia arengul kindlasti silma peal ja loodab, et aina enam veebiarendajaid leiavad enda jaoks antud tehnoloogia.

A Manual of Features of MS Silverlight Development Platform.

Interactive website creating solution Microsoft Silverlight technology has reached to its 4-th version, which was released on the 15th of April 2010. The new version provides users with some completely new opportunities and several opportunities, which were initially possible to achieve only by using Adobe Flash technology. Due to the new features the technology collects more and more fame as users discover the wide range of possibilities. Author of the current thesis has recently discovered the Silverlight technology for herself and wanted to learn more about it.

Despite the fact that technology is developing rapidly, has comfortable development tools and quite extensive possibilities, Silverlight has not been used as much as it could be. So far most of the web developers are using Adobe Flash to create interactive or animated content for websites. One of the main reasons is the lack of knowledge, because the technology options and ability to use it has not reached the potential users.

In the first chapter of the thesis the author introduces the most common version of Silverlight which is the 3-rd version. It is preferable for the user to have knowledge about the previous version of the software while learning the upgrades and new opportunities. The author also introduces 10 of the new features of Silverlight 4, which have been the most widely discussed in Silverlight forum. Additionally, the author adds one more feature that she thinks deserves more attention.

In the second chapter the author presents four of the new features that the Estonian Silverlight developers offered to be the most interesting innovations in Silverlight 4. The four of the new features are: local file access, rendering HTML in Silverlight application, audio and video local recording and finally distributing webcam and microphone signals over the network.

In the third chapter the author introduces the new opportunities and makes sample exercises for each of these features which can be used for learning. The explanations of the sample exercises can be found in the appendix. The author also describes some unexplained exercises which allow students to try the features by themselves.

In the practical part the following points can be found: creating a new Silverlight project, local file reading and writing, rendering HTML inside Silverlight application, the use of webcam and microphone and practice exercises.

Due to the rapid evolution of Silverlight technology the author is determined to keep an eye on the technology and hopes that more web developers find Silverlight useful in their work.

Kasutatud kirjandus

The Official Microsoft Silverlight Site 2009: <http://www.silverlight.net/> (Kasutamise kuupäev: 17. Veebruar 2010. a.)

The Official Microsoft Silverlight Site 2009, Silverlight 4:
<http://silverlight.net/getstarted/silverlight-4/> (Kasutamise kuupäev: 26. Märts 2010. a.)

Greń, M. 19. November 2009, *10 Best Features of Silverlight 4*:
<http://blog.goyello.com/2009/11/19/10-best-features-of-silverlight-4/> (Kasutamise kuupäev: 9. Aprill 2010. a.)

Neuback, J. 10. Detsember 2009, *Capturing the Webcam in Silverlight 4*:
<http://www.silverlightshow.net/items/Capturing-the-Webcam-in-Silverlight-4.aspx>
(Kasutamise kuupäev: 29. Märts 2010. a.)

Cleeren, G. 11. Detsember 2009, *Local File Access in Silverlight 4*:
<http://www.snowball.be/Silverlight+Advent+Calendar+December+11th+Local+File+Access+In+Silverlight+4.aspx> (Kasutamise kuupäev: 1. Aprill 2010. a.)

Cleeren, G. 10. Detsember 2009, *Working With the Webcam from Silverlight 4*:
<http://www.snowball.be/2009/12/10/Silverlight+Advent+Calendar+December+10th+Working+With+The+Webcam+From+Silverlight+4.aspx> (Kasutamise kuupäev: 1. Aprill 2010. a.)

Cleeren, G. 2. Detsember 2009, *The New Silverlight 4 WebBrowser Control*:
<http://www.snowball.be/2009/12/02/Silverlight+Advent+Calendar+December+3rd+The+New+Silverlight+4+WebBrowser+Control.aspx> (Kasutamise kuupäev: 2. Aprill 2010. a.)

Golesh, A. 18. November 2009, *Elevated Out of Browser Application*:
http://blogs.microsoft.co.il/blogs/alex_golesh/archive/2009/11/18/silverlight-4-new-features-overview-part-3-elevated-out-of-browser-applications.aspx (Kasutamise kuupäev: 2. Aprill 2010. a.)

Prosis, J. 22. Detsember 2009, *Silverlight 4's New HTML Hosting Support*:
<http://www.wintellect.com/CS/blogs/jprosis/archive/2009/12/22/silverlight-4-s-new-html-hosting-support.aspx> (Kasutamise kuupäev: 3. Aprill 2010. a.)

Abdelqader, F. 12. Märts 2010, *Your First Step to the Silverlight Voice/Video Chatting Client/Server*:
<http://www.codeproject.com/KB/IP/SilverlightVoiceVideoChat.aspx>
(Kasutamise kuupäev: 6. Aprill 2010. a.)

Yack, D. 18. November 2009, *Mixing Silverlight 4 With Dynamics CRM*:
<http://crm.davidyack.com/journal/2009/11/18/mixing-silverlight-4-with-dynamics-crm.html>
(Kasutamise kuupäev: 7. Aprill 2010. a.)

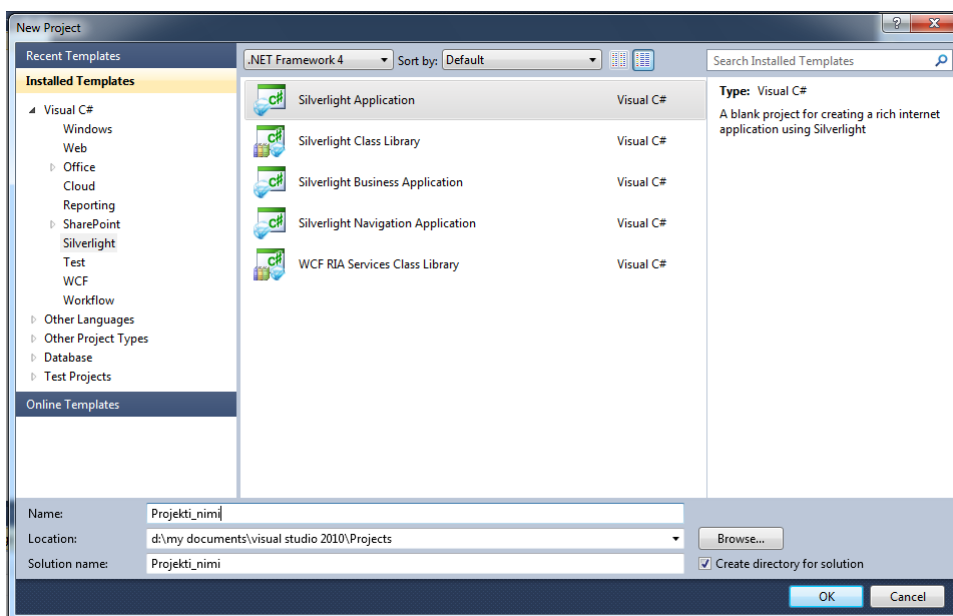
Lühendite seletused

| | | |
|-------------|---|---|
| RC | <i>Release Candidate</i> | Testversioon |
| AAC | <i>Advanced Audio Coding</i> | Edasijõudnud audio dekodeerimine |
| HDTV | <i>High-Definition Television</i> | Kõrgresolutsiooniga televisioon |
| GPU | <i>Graphics Processing Unit</i> | Graafikaprotsessor |
| UI | <i>User Interface</i> | Kasutajaliides |
| API | <i>Application Programming Interface</i> | Rakendusliides |
| COM | <i>Component Object Model</i> | Komponentobjektide mudel |
| WCF | <i>Windows Communication Foundation</i> | Windowsi Kommunikatsiooni Fond |
| HTML | <i>Hypertext Markup Language</i> | Hüperteksti märkimiskeel |
| USB | <i>Universal Serial Bus</i> | Universaalne järjestiksiin |
| RIA | <i>Rich Internet Application</i> | Rikkasisuline veebirakendus |
| CLR | <i>Common Language Runtime</i> | Keeleülene käitusajamoodul |
| OOB | <i>Out of Browser</i> | Brauseri väline |
| URI | <i>Uniform Resource Identifier</i> | Ühtne ressursi-identifikaator |
| PDC | <i>Professional Developers Conference</i> | Professionaalsete arendajate konverents |
| RTP | <i>Real-time Transport Protocol</i> | Reaalaja transpordiprotokoll |
| TCP | <i>Transmission Control Protocol</i> | Edastusohje protokollistik |
| UDP | <i>User Datagram Protocol</i> | Kasutajadatagrammi protokoll |
| HD | <i>High-Definition</i> | Kõrgresolutsioon |

LISAD

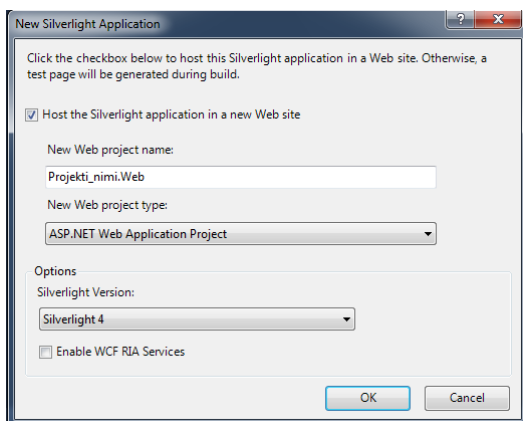
Lisa 1. Uue Silverlight projekti loomine

Uue projekti loomiseks tuleb avada Microsoft Visual Studio 2010, esmakordsel avamisel on võimalik valida eelistatum programmeerimiskeel, millele järgneb programmi valitud keele seadmete laadimine, antud juhul tuleks valida C#. Seejärel tuleb valida menüüst *File*→*New*→*Project...*→*Silverlight Application* (vt. Joonis 2).



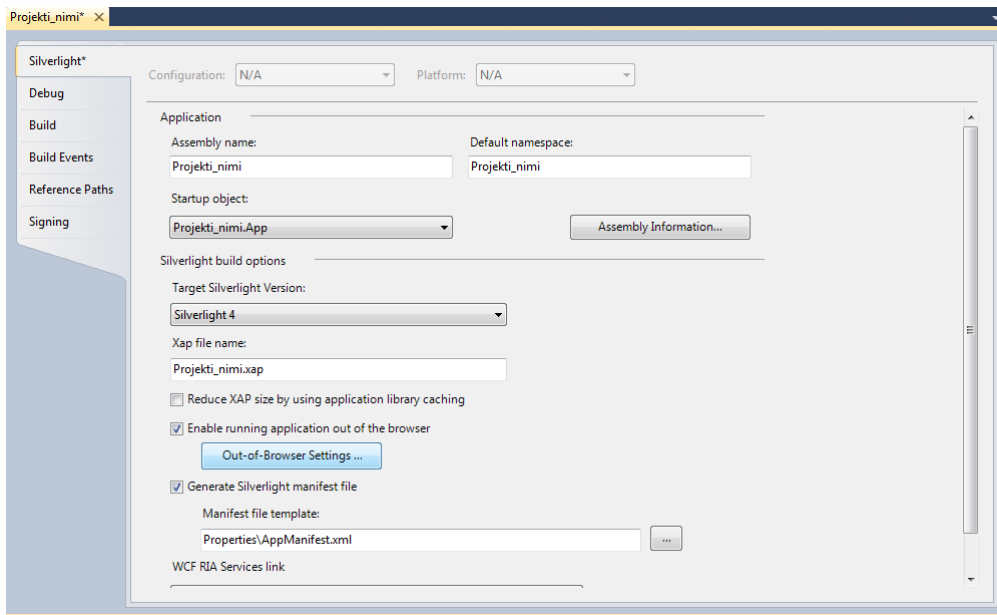
Joonis 2. Uue projekti loomine

Nüüd on võimalik määrata projektile soovitud nimi (antud juhul on nimeks: Projekti_nimi). Klõpsata nupule OK, mille järel avaneb uus aken, mis võimaldab valida Silverlight'i versiooni ning näitab täpsustusi rakenduse ja ASP.NET lehe sidumise kohta (vt. Joonis 3).



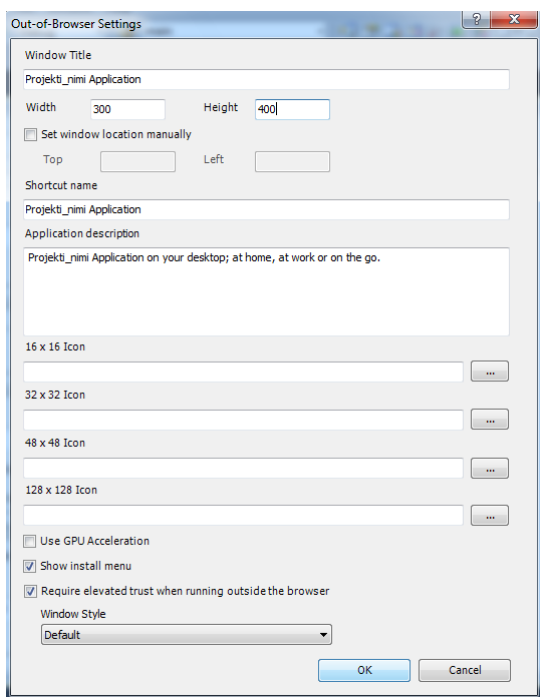
Joonis 3. Silverlight'i täpsustused

Selleks, et oleks võimalik koostada brauseri välist (OOB) rakendust, tuleb menüüst avada *Project*→*Projekti_nimi Properties...*, seejärel avaneb rakenduse atribuutide aken (vt Joonis 4)



Joonis 4. Silverlight rakenduse atribuudid

Selleks, et rakendus oleks brauseriväline, tuleb ära märkida valik, millele järgneb tekst *Enable running application out of the browser*. Seejärel tuleb avada veebivälise rakenduse atribuudid - *Out-of-Browser Settings...* ja märkida ära valik, mis kohustab kasutajal andma kõrgendatud õigused rakenduse käivitamisel – *Require elevated trust when running outside the browser*. Samuti on seal võimalik ära määrata rakenduse erinevad mõõtmed. (vt. Joonis 5)



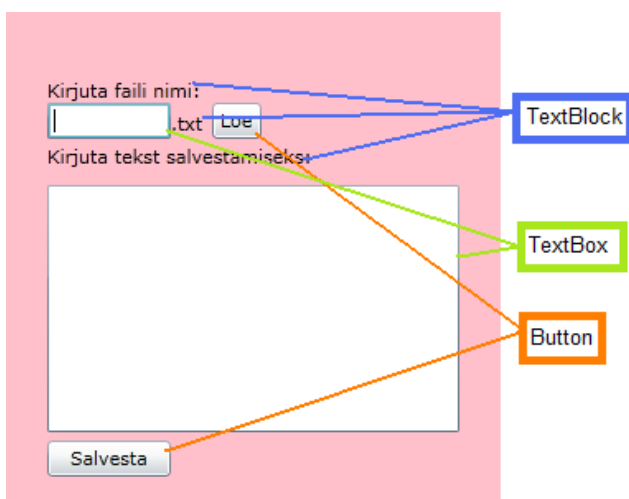
Joonis 5. Veebivälise rakenduse atribuudid

Lisa 2. Kohalikult kettalt lugemine ja kirjutamine

Rakenduse eesmärgiks on demonstreerida liivakasti välist rakendust, mis võimaldab lugeda tekstifaili kohalikult kettalt ning salvestada tekstiväljale kirjutatud teksti. Samuti tutvustab rakendus võimalust pääseda kindlatesse kataloogidesse, kuhu varem oli juurdepääs keelatud, et tagada turvalisus.

Esmalt loob autor rakenduse, millega on võimalik lugeda kohalikult kettalt näiteks tekstifaili ja salvestada tekstiväljale kirjutatud jutt tekstifaili, millele on võimalik ise nimi määrata.

Esmalt tuleb kas avada või luua uus (*Project*→*Add New Item*→*Silverlight User Control*) XAML fail, antud juhul kasutas autor projekti loomisel automaatselt tekkinud *MainPage.xaml* faili. Algatuseks tuleb kujundada rakendusele kasutajaliides, mis on iga arendaja oma teha. Järgnevalt toob autor välja enda loodud rakenduse kujunduse (vt. Joonis 6).



Joonis 6. Rakenduse kasutajaliidese objektide paigutus

Kasutajaliidese põhjaks on *Canvas* tüüpi paigaldushaldur, mille sees on võimalik erinevad objektid tööriistakastist (*Toolbox*) lohistada disaini vaatesse ja meelepäraselt paigutada. Peale erinevate objektide lohistamist ja paigutamist tööpinna tekib antud juhul *MainPage.xaml* faili automaatselt kood, mis kirjeldab antud rakenduse kasutajaliidese. (vt. Koodinäide 1)

```
<Canvas x:Name="Juur" Background="Pink">
  <Button Canvas.Left="25" Canvas.Top="15" x:Name="paigalda_nupp"
    Content="Paigalada rakendus" Height="22" Width="125"
    Click="paigalda_nupp_Click" />
  <TextBlock Canvas.Left="25" Canvas.Top="40" x:Name="textBlock1"
    Text="Kirjuta faili nimi:" Height="16" Width="100" />
  <TextBox Canvas.Left="25" Canvas.Top="55" x:Name="failinimekast"
    Height="22" Width="75" />
  <TextBlock Canvas.Left="100" Canvas.Top="60" x:Name="txt" Text=".txt"
    Height="15" Width="25" />
  <Button Canvas.Left="125" Canvas.Top="55" x:Name="loe_nupp" Content="Loe"
    Height="22" Width="30" Click="loe_nupp_Click" />
```

```

<TextBlock Canvas.Left="25" Canvas.Top="80" x:Name="textBlock2"
    Text="Kirjuta tekst salvestamiseks:" Height="16" Width="175" />
<TextBox Canvas.Left="-5" Canvas.Top="75" x:Name="tekst" Height="150"
    Width="250" Margin="30,30,0,0" HorizontalAlignment="Left"
    VerticalAlignment="Top" />
<Button Canvas.Left="25" Canvas.Top="260" x:Name="salvesta_nupp"
    Content="Salvesta" Height="22" Width="75" Click="salvesta_nupp_Click" />
</Canvas>

```

Koodinäide 1. MainPage.xaml

Nagu juba eelpool mainitud on faili põhjaks *Canvas* tüüpi paigaldushaldur, mis sai nimeks Juur ja mille tausta värviks valis autor roosa. *Canvas*'e sees olevate objektide asukohad on ära määratud x ja y koordinaatide abil (*Canvas.Left* ja *Canvas.Top*). Kõikidele objektidele on määratud käsklusega *x:Name* nimi, mille kaudu on hiljem võimalik objektideni jõuda. Selleks, et nupu vajutamisel ka midagi toimuks, tuleb lisada *Click* sündmus (*Event*) ning programm pakub välja, et loob kasutaja asemel uue sündmuse (*New Event Handler*) *MainPage.xaml.cs* faili, milles asub nn disaini taustal olev kood (*Code Behind File*), kus on sündmuste toimumise kood, mida ei ole kasutajale näha. Sama moodi on ühele tekstiväljadest lisatud sündmus, mis aktiveerub kui vajutada *Enter* klahvi.

Selleks, et rakendus töötaks ja võimaldaks kasutajale meelepäraseid funktsioone ja sündmuste kirjeldusi, on tarvis *xaml.cs* failis välja kutsuda ja luua lisaks automaatselt genereeritavatele teekidele uued teegid, mis seda võimaldavad. Antud projekti jaoks on lisaks vaja vaid ühte teeki, mille abil on võimalik failisüsteemi salvestada (vt. Koodinäide 2).

```
using System.IO;
```

Koodinäide 2. Uus failisüsteemi salvestamist võimaldav teek

Nagu eelpool mainitud, genereerib programm erinevate sündmuste defineerimiseks automaatselt koodi *MainPage.xaml.cs* faili, seega tuleb ise käsitsi kirjutades lisada objektide sündmustele vaid meelepärased käsklused. Järgnevalt toob autor välja käsil oleva rakenduse erinevate objektide sündmuste kirjeldused.

Kuna antud rakendus töötab vaid brauseri välisena, kuna kohalikule kettale juurdepääs vajab kõrgendatud õigusi, tuleb kasutajal rakendus arvutisse alla laadida. Selleks on kaks võimalust:

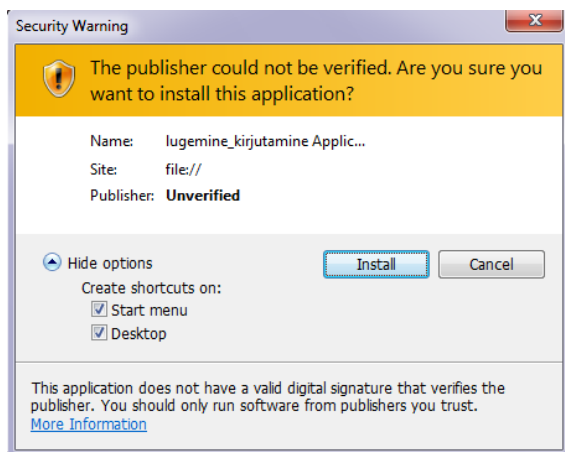
- veebilehel paremale hiireklahvile klõpsamisel antakse kasutajale võimalus antud rakendus alla laadida
- kasutajaliidesesse luuakse nupp, mis antud juhul sai nimeks *paigalda_nupp* ning kuna XAML failis lisas autor nupule sündmuse *Click* siis *xaml.cs* failis on võimalik antud funktsioonile lisada käsk, mis laeb antud rakenduse alla (vt. Koodinäide 3).

```
private void paigalda_nupp_Click(object sender, RoutedEventArgs e)
{
    App.Current.Install();
}

```

Koodinäide 3. Rakenduse paigaldamise nupu funktsiooni kood

Peale seda, kui kasutaja on avaldanud soovi rakenduse paigaldamiseks, peab kasutaja andma oma nõusoleku kõrgendatud õiguste päringu dialoogis ning määrama rakendusele asukoha (*Desktop* või *Start menu*). (vt. Joonis 7)



Joonis 7. Kõrgendatud õiguste päringu dialoog

Selleks, et paigaldamise nupp peale rakenduse alla laadimist enam nähtaval ei oleks, tuleb lisada vastav käsk, mis võimaldab nupu ära peita (vt Koodinäide 4).

```
public MainPage()
{
    InitializeComponent();
    //Kontrollime, kas rakendus on installeeritud ning
    //kui on siis peidame rakenduse paigaldamise nupu.
    if (App.Current.InstallState == InstallState.Installed)
    {
        paigalda_nupp.Visibility = Visibility.Collapsed;
    }
}

```

Koodinäide 4. Paigaldamise nupu peitmine

Samamoodi nagu paigaldamise nupule lisas autor ka lugemise ja salvestamise nupule sündmuse *Click*. Selleks, et rakendus saaks kasutaja kohalikult kettalt valitud kataloogist (Dokumendi, Muusika, Videod või Pildid) lugeda mingisugust olemasolevat faili, tuleb kasutada funktsiooni *Environment.SpecialFolder*. Antud funktsioon võimaldab leida arvutist valitud kataloog ning seal sees olevad failid, antud juhul on kataloogiks Dokumendid. Kasutaja poolt sisestatud nime järgi otsitakse antud fail ning kuvatakse selle sisu teatekastis (*MessageBox*), kui aga faili ei eksisteeri, siis teatatakse sellest kasutajale samuti teatekastis (vt. Koodinäide 5).

```

private void loe_nupp_Click(object sender, RoutedEventArgs e)
{
    string nimi= failinimekast.Text.ToString();
    string failinimi = Environment.GetFolderPath
        (Environment.SpecialFolder.MyDocuments) + "/" + nimi+".txt";
    if (File.Exists(failinimi))
    {
        MessageBox.Show(File.ReadAllText(failinimi));
    }
    else
    {
        MessageBox.Show("Faili: " + (failinimi) + " ei eksisteeri.");
    }
}

```

Koodinäide 5. Faili lugemine kohalikult kettalt

Faili salvestamine kohalikule kettale sarnaneb vägagi lugemisega. Kasutajal on võimalik anda loodavale failile nimi, mille ta sisestab esimesse tekstikasti ning teise tekstikasti tuleb kirjutada tekst, mida soovitakse faili salvestada. Seejärel peale salvestamise nupu vajutamist loob rakendus faili ning teatab kasutajale, et fail on salvestatud (vt. Koodinäide 6).

```

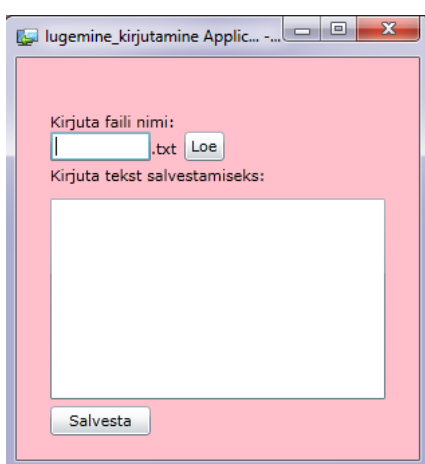
private void salvesta_nupp_Click(object sender, RoutedEventArgs e)
{
    string nimi = failinimekast.Text.ToString();
    string failinimi = Environment.GetFolderPath
        (Environment.SpecialFolder.MyDocuments) + "/" + nimi+".txt";
    string salvestustekst = tekst.Text;
    File.WriteAllText(failinimi, salvestustekst);
    MessageBox.Show("Fail " + failinimi + " on salvestatud.");
}

```

Koodinäide 6. Failide salvestamine kohalikule kettale

Sarnaselt tekstifailidele on võimalik lugeda ja kirjutada ka teisi failivorminguid nagu näiteks DOC, XLS jne.

Lõplik veebivälise rakenduse välimust vaata Joonis 8-lt.

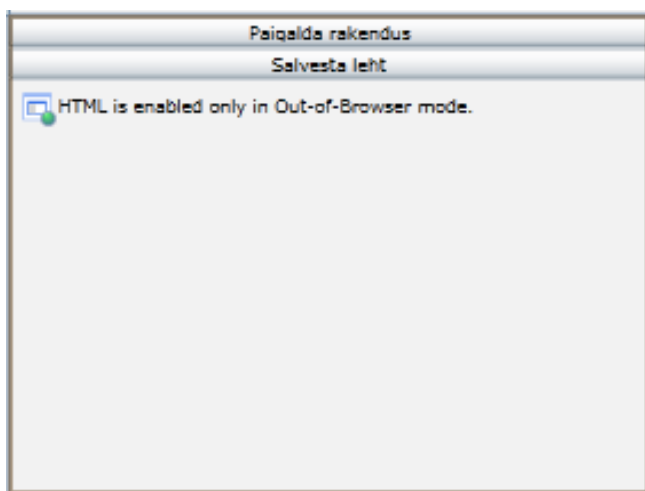


Joonis 8. Faili lugemise ja salvestamise lõpliku rakenduse välimust

Lisa 3. HTML sisu kuvamine Silverlight rakenduse sees

Järgneva rakenduse eesmärk on tutvustada lugejale veebilehe juhtelemendi (*WebBrowser Control*) ja veebibrauseripintsi (*WebBrowserBrush*) kasutamist, mis võimaldavad rakenduse sisse kuvada veebilehe HTML elemendi, kus on võimalik ringi liikuda ning tekstiväljale kirjutada aadress, millisele veebilehele soovitakse järgmisena jõuda.

Esimese asjana tuleb luua kas uus projekt või olemasolevasse projekti luua uus xaml fail. Ka see on sarnaselt eelnevale näitele brauseriväline rakendus, kuna HTML'i ei ole võimalik kuvada veebikeskkonnas paiknevas Silverlight'i rakenduses. Selle rakenduse loomisel tegi autor uue projekti ning alustas rakenduse loomist MainPage.xaml faili. Rakenduse kasutajaliides on lihtsa kujundusega selle põhjaks on *Grid* tüüpi paigaldushaldur, mis koosneb kolmest reast. Esimesel real asuvad tekstiväli ja rakenduse paigaldamise nupp, teisel real on nupp, mis salvestab kohalikule kettale kas algset rakendusel avanevat lehte või siis lehe, mille täpne aadress sisestatakse tekstiväljale. Kolmandal real asub veebibrauseri juhtelement (*WebBrowser Control*), mille abil siis kuvatakse rakenduses HTML sisu (vt Joonis 9).



Joonis 9. HTML sisu Silverlight rakenduse sees kasutajaliides

Kasutajaliidese XAML kood on suhteliselt lühike ning arusaadav. Esmalt tuleb määrata *Grid*'i ridadele täpsedkõrgused (*Grid.RowDefinitions*) ning seejärel lisada ridadele soovitud objektid. Selle rakenduse objektideks on 2 nuppu ja tekstiväli, millele on vaja lisada sündmused, et hiljem oleks võimalik sisestada täpsustused faili taga asuvasse xaml.cs koodi. Objektidele on asukohad võimalik määratleda *Grid.Row* käsklusega (vt. Koodinäide 7).

```
<Grid x:Name="Juur">  
  <Grid.RowDefinitions>  
    <RowDefinition Height="20"></RowDefinition>  
    <RowDefinition Height="20"></RowDefinition>
```

```

        <RowDefinition></RowDefinition>
    </Grid.RowDefinitions>
    <TextBox x:Name="uusaaddress" Grid.Row="0" KeyDown="uusaaddress_KeyDown" />
    <Button x:Name="paigalda_nupp" Content="Paigalda rakendus" Grid.Row="0"
        Click="paigalda_nupp_Click" />
    <Button x:Name="salvesta_nupp" Content="Salvesta leht" Grid.Row="1"
        Click="salvesta_nupp_Click" />
    <WebBrowser x:Name="sirvik" Source="http://www.tlu.ee" Grid.Row="2" />
</Grid>

```

Koodinäide 7. Kasutajaliideses asuva *Grid*'i kirjeldus

Sarnaselt eelmisele rakendusele on ka siin xaml.cs failis vaja lisaks ühte teeki, mis võimaldab failisüsteemi salvestada. (vt Koodinäide 8).

```
using System.IO;
```

Koodinäide 8. Failisüsteemi salvestamise teek

Samamoodi, nagu esimeses rakenduse näites, on määratud ära paigaldamise nupu funktsioonid ning nupu peitmine juhul, kui rakendus on juba kasutaja arvutisse alla laetud. Seejärel lisada tekstivälja sündmusele, milleks on *Enter* klahvi vajutamine. See on täpsustus, mis aktiveerumisel kuvab veebibrauseri juhtelementi veebikeskkonna, mille aadress on sisestatud tekstiväljale (vt. Koodinäide 9).

```

private void uusaaddress_KeyDown(object sender, KeyEventArgs e)
{
    if (e.Key == Key.Enter)
    {
        sirvik.Navigate(new Uri("http://" + uusaaddress.Text));
    }
}

```

Koodinäide 9. Veebilehe navigeerimine tekstiväljale sisestatud aadressile

Salvestamise nupp salvestab kahjuks ainult algse või tekstiväljale sisestatud aadressiga veebilehe, kuna Silverlight 4 ei võimalda rakendusel esitada päringut hetkel aktiivse URI saamiseks. Esmalt esitatakse päring veebibrauseri juhtelemendile URI aadressi saamiseks, seejärel tuleb luua veebiklient, mis loob uue funktsiooni alla laadimise sündmuse (*new DownloadStringCompletedEventHandler*) kirjeldamiseks, mille abil on võimalik hiljem veebileht salvestada kohalikule kettale (vt. Koodinäide 10).

```

private void salvesta_nupp_Click(object sender, RoutedEventArgs e)
{
    Uri address = sirvik.Source;
    WebClient klient = new WebClient();
    klient.DownloadStringCompleted += new DownloadStringCompletedEventHandler
        (klient.DownloadStringCompleted);
    klient.DownloadStringAsync(address);
}

```

Koodinäide 10. Lehe salvestamise nupu täpsustused

Eelpool mainitud uue sündmusele kirjeldusega on võimalik veebileht salvestada kohalikule kettale. Nüüd võib tekkida paljudel küsimus, et milleks salvestada veebilehtesid, kui on

olemas internetiühendus. Kuid autoril oli algselt plaanitud salvestada näiteks erinevaid veebis pakutavaid toiduretsepte, mida saaks hiljem kasvõi maakodus vanaemale näidata või ise kasutada kuskil, kus ei ole interneti ühendust (vt. Koodinäide 11).

```
void klient_DownloadStringCompleted(object sender,
    DownloadStringCompletedEventArgs e)
{
    File.WriteAllText(Environment.GetFolderPath
        (Environment.SpecialFolder.MyDocuments) + "/" + uusaadress.Text + ".html",
        e.Result);
    MessageBox.Show("Leht on salvestatud");
}
```

Koodinäide 11. Uus lehe alla laadimise sündmuse kirjeldus

Selle rakenduse veebiväline väljanägemine on näha järgneval joonisel (vt. Joonis 10).

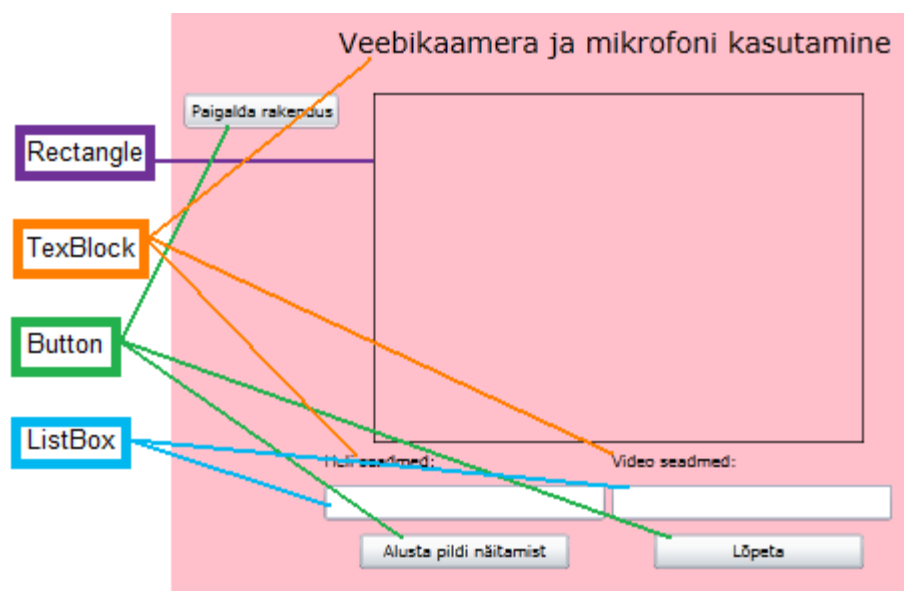


Joonis 10. Veebibrauseri välise rakenduse kujundus

Lisa 4. Veebikaamera ja mikrofoni kasutamine

Järgneva rakenduse eesmärgiks on tutvustada lugejale Silverlight 4-a uut, huvitavat ja arendajate poolt kauaoodatud võimalust kasutada kohaliku arvutiga ühendatud veebikaamerat ja mikrofoni.

Samamoodi nagu eelmistes näidisrakendustes, tuleks ka siin luua kas uus projekt või xaml fail, autor tegi antud juhul uue projekti ning taas alustas rakenduse loomist kasutajaliidestest, mis on kirjeldatud MainPage.xaml failis. Kasutajaliidese põhjaks on sarnaselt esimesele näidisrakendusele *Canvas* tüüpi paigutushaldur, millele on lohistatud meelepärased objektid. Selle rakenduse juures on objektideks tekstiplokid (*TextBlock*), ristkülik (*Rectangle*), nimekirjakastid (*ListBox*) ja nupud (*Button*) (vt. Joonis 11).



Joonis 11. Kasutajaliidesele paiknevad objektid

Rakenduses on kaks nimekirjakasti (*ListBox*), kus ühte kuvab rakendus kõik arvutiga ühendatud heli seadmed ja teise video seadmed. Peale soovitud seadmete märgistamist kuvab rakendus veebikaamera otsepildi ristküliku sisse, kasutades videopintslit (*VideoBrush*). „Alusta_nupp“ käivitab seadmed ning „Lõpeta_nupp“ peatab kõik seadmed. Autor kujundas MainPage.xaml väljanägemise disaini vaates ning tänu sellele on kood genereeritud Visual Studio 2010 poolt (vt. Koodinäide 12).

```
<Canvas x:Name="Juur" Background="Pink">
  <TextBlock Canvas.Left="125" Canvas.Top="10" x:Name="pealkiri"
    Text="Veebikaamera ja mikrofoni kasutamine" Height="50" Width="400"
    FontSize="20" />
  <Rectangle Canvas.Left="150" Canvas.Top="60" x:Name="pildikast" Stroke="Black"
    StrokeThickness="1" Height="250" Width="350" />
  <Button Canvas.Left="-135" Canvas.Top="0" x:Name="paigalda_nupp"
```

```

        Content="Paigalda rakendus" Margin="150,60,0,0" Height="25" Width="110"
        Click="paigalda_nupp_Click" ></Button>
<ListBox Canvas.Left="115" Canvas.Top="340" x:Name="helikast" Height="25"
    Width="200" >
    <ListBox.ItemTemplate>
        <DataTemplate>
            <TextBlock Text="{Binding FriendlyName}"/>
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>
<TextBlock Canvas.Left="115" Canvas.Top="315" x:Name="heliblokk"
    Text="Heli seadmed:" Height="25" Width="200" />
<ListBox Canvas.Left="320" Canvas.Top="340" x:Name="videokast" Height="25"
    Width="200" >
    <ListBox.ItemTemplate>
        <DataTemplate>
            <TextBlock Text="{Binding FriendlyName}"/>
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>
<TextBlock Canvas.Left="320" Canvas.Top="315" x:Name="videoblokk"
    Text="Video seadmed:" Height="25" Width="200" />
<Button Canvas.Left="140" Canvas.Top="375" Content="Alusta pildi näitamist"
    x:Name="alusta_nupp" Height="25" Width="150" Click="alusta_nupp_Click"/>
<Button Canvas.Left="350" Canvas.Top="375" Content="Lõpeta" x:Name="lõpeta_nupp"
    Height="25" Width="150" Click="lõpeta_nupp_Click"/>
</Canvas>

```

Koodinäide 12. Kasutajaliidese kujundus

Antud rakenduse xaml.cs faili ei ole tarvis juurde lisada teeke, kuna kõik vajaminevad teegid on juba automaatselt genereeritud.

Samamoodi nagu eelnevates näidiskodudes, tuleb ära määrata paigaldamise nupu funktsioonid ja nupu peitmine, siis kui rakendus on juba kohalikku arvutisse laetud. Samas töötavad veebikaamera ja mikrofoni väga edukalt ka veebis, seega ei ole rakenduse paigaldamine kohustuslik, kuid kui on soov salvestada, siis tuleb lisada failisüsteemi salvestamise teek (System.IO) ja paigaldada rakendus arvutisse.

Selleks, et leida arvutiga ühendatud veebikaamera(d) ja mikrofoni(id), tuleb esmalt defineerida seadmehõive objekt (*CaptureSource*), mis hiljem kuvab video ja heli. Seejärel tuleb luua uus suunamise sündmus (*New RoutedEventHandler*). Uue *MainPage_Loaded* sündmuse kirjeldus tuleks lisada sarnaselt paigaldamise nupule *public MainPage()*-i. (vt. Koodinäide 13)

```

CaptureSource cs;
public MainPage()
{
    InitializeComponent();
    if (App.Current.InstallState == InstallState.Installed)
    {
        paigalda_nupp.Visibility = Visibility.Collapsed;
    }
    Loaded += new RoutedEventHandler(MainPage_Loaded);
}

```

Koodinäide 13. Uue *MainPage_Loaded* sündmuse loomine

MainPage_Loaded sündmuse kaudu on võimalik kuvada nimekirjakastidesse (*ListBox*) arvutiga ühendatud heli ja video seadmete nimekirjad ning need vastavatesse kastidesse paigutada. Samuti tuleb luua uus seadmehõive objekt, mis hiljem püüab seadmetest video ja heli (*CaptureSource*). (vt. Koodinäide 14)

```
void MainPage_Loaded(object sender, RoutedEventArgs e)
{
    //heli seadmete nimekiri
    helikast.ItemsSource = CaptureDeviceConfiguration.
        GetAvailableAudioCaptureDevices();
    //video seadmete nimekiri
    videokast.ItemsSource = CaptureDeviceConfiguration.
        GetAvailableVideoCaptureDevices();
    //seadmehõive loomine
    cs = new CaptureSource();
}
```

Koodinäide 14. *MainPage_Loaded* sündmuse kirjeldus

Seejärel on võimalik ära kirjeldada, mis juhtub alustamise nupule vajutades. Esmalt tuleb kontrollida, et seadmehõive objekt (*CaptureSource*) ei oleks tühi. Seejärel tuleb peatada kõik seadmed, kui juhuslikult peaks mingi seade olema aktiivne. Järgmiseks tuleb aktiveerida seadmed, mis kasutaja on esitatud nimekirjast valinud ning siis on võimalik risküliku sisu videopintsliga (*VideoBrush*) abiga katta. Nüüd on vaja veel kasutajalt küsida luba seadmete kasutamiseks ning kui kasutaja annab oma loa, alustab rakendus oma tööd ning kuvab riskülikusse otsepildi veebikaamerast. (vt. Koodinäide 15)

```
private void alusta_nupp_Click(object sender, RoutedEventArgs e)
{
    if (cs != null)
    {
        //peatame kõik seadmed, kui peaks midagi käima
        cs.Stop();
        //rakendus seab uued seadmed mis on listboxist valitud
        cs.VideoCaptureDevice = (VideoCaptureDevice)videokast.SelectedItem;
        cs.AudioCaptureDevice = (AudioCaptureDevice)helikast.SelectedItem;
        //Loomme videopintslit, et täitsa riskülik videopildiga
        VideoBrush videopintsel = new VideoBrush();
        videopintsel.SetSource(cs);
        pildikast.Fill = videopintsel;
        //küsime seadmete kasutamiseks luba
        if (CaptureDeviceConfiguration.AllowedDeviceAccess ||
            CaptureDeviceConfiguration.RequestDeviceAccess())
        {
            cs.Start();
        }
    }
}
```

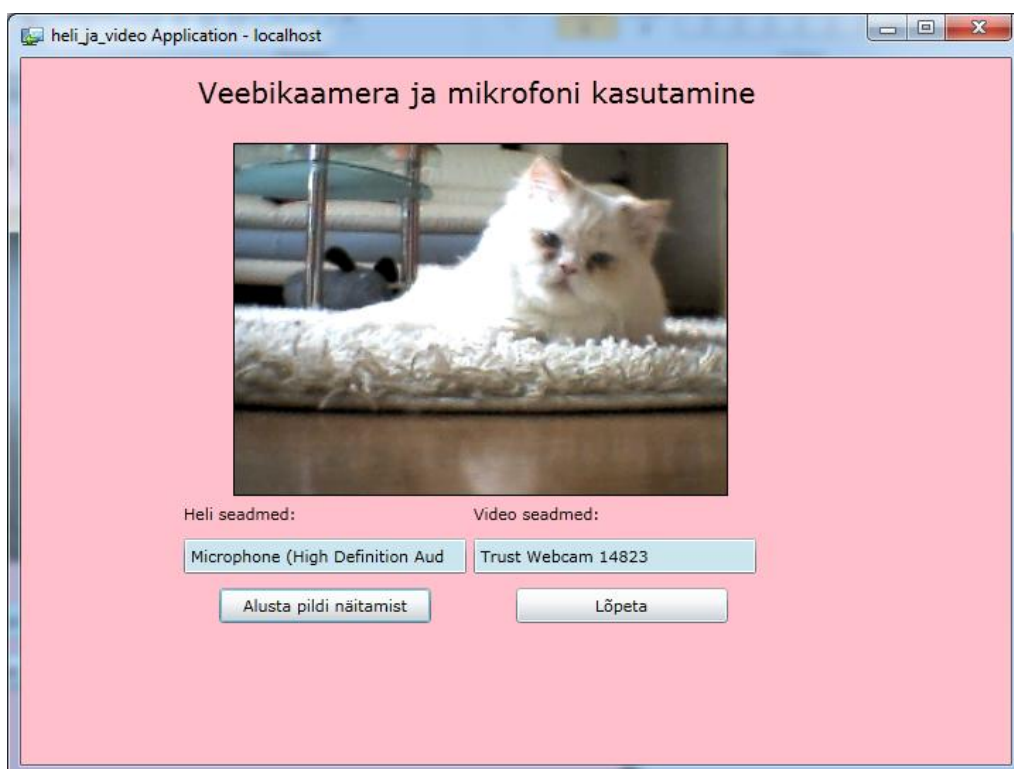
Koodinäide 15. Seadmetest info püüdmise alustamise nupu kirjeldus

Lõpetamise nupu kirjeldus näeb välja lihtne ja arusaadav. Esmalt tuleb kontrollida, et oleks olemas seadmehõive objekt (*CaptureSource*) ning seejärel tuleb peatada kõikide seadmete töö. (vt. Koodinäide 16)

```
private void l6peta_nupp_Click(object sender, RoutedEventArgs e)
{
    if (cs != null)
    {
        cs.Stop();
    }
}
```

Koodinäide 16. Seadmetest info püüdmise lõpetamise nupu kirjeldus

Valmis rakenduse kasutajaliides näeb välja järgnev (vt. Joonis 12):



Joonis 12. Veebikaamera ja mikrofoni rakenduse kasutajaliides

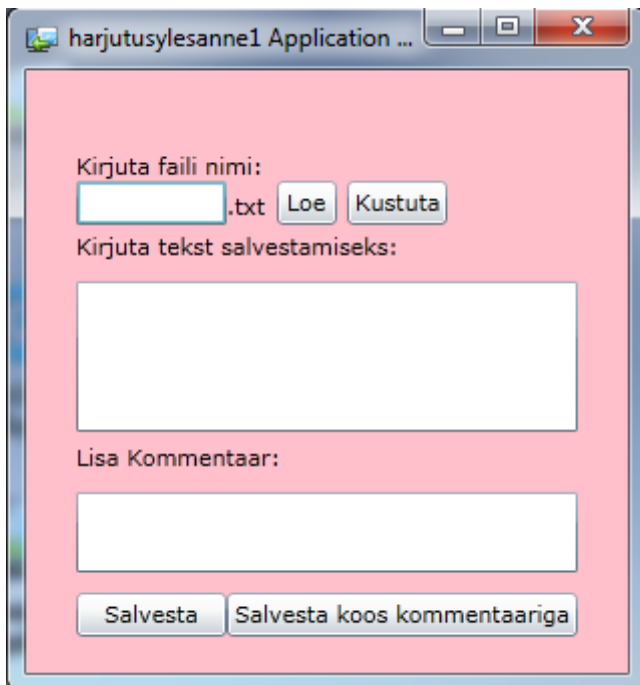
Lisa 5. Harjutusülesanded

Järgnevalt pakub autor välja mõned lihtsamad harjutusülesanded eelnevate rakenduste täiendamiseks, et õppuritel oleks võimalik ka omal käel uuendusi proovida. Autori rakendused ja lähtekoodid kõikide rakenduste kohta asuvad CD-l.

a. Kohalikult kettalt lugemine ja kirjutamine

- Loo Dokumentide kataloogi uus kataloog. Seejärel luba kasutajal salvesta antud kataloogi tekstifail ning peale salvestamist võimalda ka selle faili lugemist.
- Lisa kasutajaliidesesse kustutamise nupp, mis leiab kustutava faili nime esimeselt tekstiväljalt ning kustutab faili ja annab sellest kasutajale teada.
- Lisa uus tekstikast, kuhu oleks võimalik kirjutada kommentaar. Samuti lisa uus salvestamise nupp, mis võimaldab salvestada teksti koos kommentaariga, lisades kommentaari ette 2 tühja rida.

Lõpplahenduse kasutajaliides näeb välja järgnev (vt. Joonis 13):

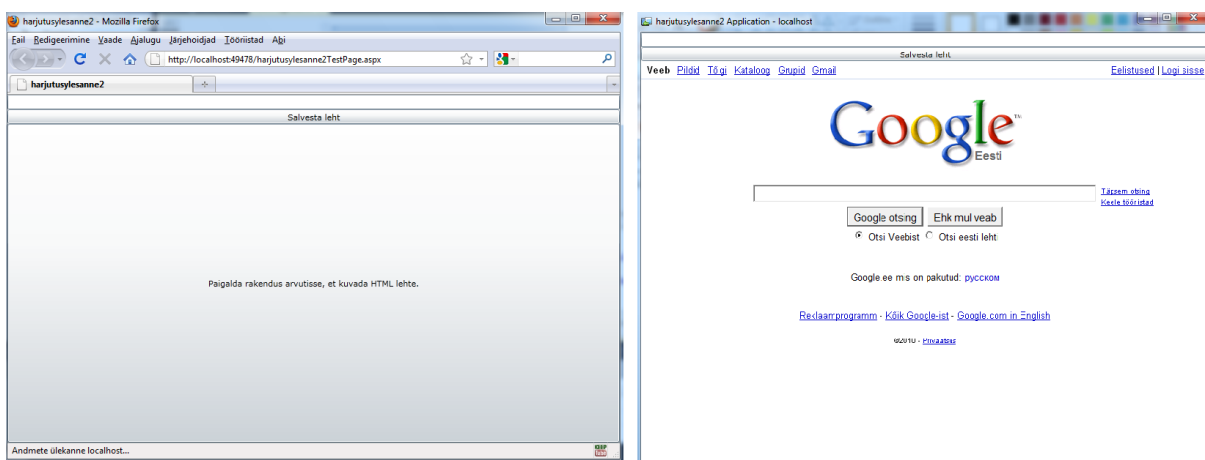


Joonis 13. Täiendatud kohalikult kettalt lugemise ja salvestamise rakendus

b. HTML sisu kuvamine Silverlight rakenduse sees

- Muuda ära rakenduses kasutatav veebileht näiteks nii, et käivitumisel avaneks *Google* leht.
- Paiguta ümber paigaldamise nupp nii, et see paikneks veebikeskkonnas veebibrauseri juhtelemendi (*WebBrowser Control*) peal ja samamoodi nagu näidisrakenduses peidetaks pärast rakenduse paigaldamist ära.

Lõpplahendused kasutajaliidesest nii veebikeskkonnas kui brauserivälises keskkonnas näevad välja järgmised (vt. Joonis 14):

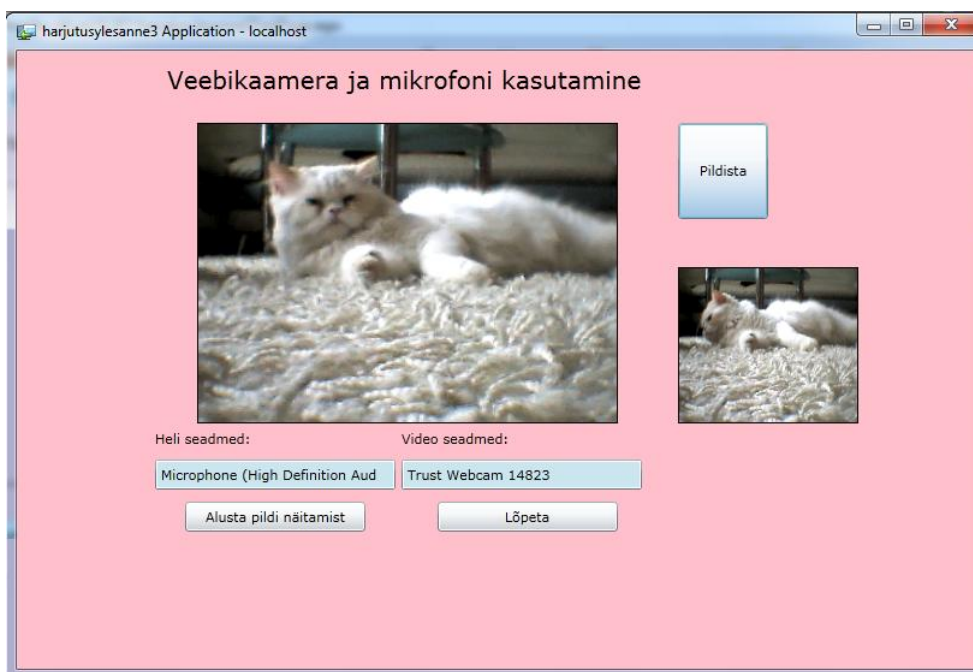


Joonis 14. HTML harjutusülesande käigus uuenenud kasutajaliides

c. Veebikaamera ja mikrofoni kasutamine

- Lisa kasutajaliidesesse pildistamise nupp ning riskülik, kuhu hiljem kuvab rakendus pildi.
 - Xsml.cs faili lisa pildipintsel (*ImageBrush*).
 - Pildipintsel loo pildistamise nupu seletuse juurde ning täida riskülik pildipintsliga.
 - Loo uus *CaptureImageCompleted* sündmus ning paiguta sündmuse tulemus pildipintslile väärtuseks.
 - Lõpetuseks kasuta *CaptureImageAsync* käsklust, et siduda saadud pilt riskülikuga.

Lõpplahendus uuendustega kasutajaliidesest näeb välja järgnev (vt. Joonis 15):



Joonis 15. Veebikaamera ja mikrofoni kasutamise rakenduse uus kasutajaliides