

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Alexander Boyko 179732IABB  
Deniss Šutov 179230IABB

**MEESKONNAPROJEKT: OLEMASOLEVA  
RAKENDUSE TEISALDAMINE *UWP*  
PLATVORMILE**

Bakalaureusetöö

Juhendajad: Jekaterina Tšukrejeva,  
Kristina Murtazin

Tallinn 2020

## **Autorideklaratsioon**

Kinnitame, et oleme koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autorid: Alexander Boyko, Deniss Šutov

17.05.2020

## Annotatsioon

Antud projekti peamine eesmärk on olemasoleva rakenduse taasloomine uuel *Windows 10* platvormil.

Rakenduse kasutajad on haigla töötajad, kes rakenduse abil saavad ligipääsu patsientide ja uuringute andmetele. Samuti rakenduse abil saab uuringu meedia failid redigeerida, näiteks joonistada uuringu jooksul tehtud piltide peal selleks, et teha märkmed.

Olemasolev rakendus on kasutusel juba kaua aega ega vasta kasutajate ootustele. Näiteks mõned kasutajaliidese disaini elemendid on juba vananenud ning vajavad uuendamist.

Probleemi lahendamiseks oli uuritud olemasoleva rakenduse funktsionalsus ning välja töötatud vajalikud nõuded uue projekti arendamiseks. Uuele rakendusele samm sammult oli lisatud uuendatud elemendid. Samuti kogu projekti arenduse käigus toimusid koosolekud ettevõttega, kus oli esitatud vahetulemused ning saadud tagasiside tehtud muudatuste kohta.

Tulemusena oli saadud taasloodud rakendus uuel platvormil olemasoleva funktsionaalsusega ja uuendatud kasutajaliidese disainiga. Oli ka lisatud vajalik funktsionaalsus juurde.

Antud töö analüüsi tulemustest selgus, et uus platvorm on väga võimekas ja rohkem adapteeritud uutele arvutitele. Aga nagu iga uus asi see vajab uurimist.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 40 leheküljel, 5 peatükki, 35 joonist, 0 tabelit.

## **Abstract**

### **Team project: Moving an existing application to UWP platform**

The main goal of this project is to recreate the original application on a new Windows 10 platform.

Users of the application are hospital staff who can access patient and study data. You can also use the application to edit the study media files, such as drawing on images taken during the study to make notes.

The main problem of the original application is that it is technologically outdated and does not meet user expectations. For example, some UI design elements are already obsolete and need to be updated.

To solve this problem, it was necessary to get acquainted with the capabilities of the original application and determine the functional and non-functional requirements for the project. A new application was created, where updated elements of the original were gradually added. As the development progressed, the application was tested for compliance with established requirements. During the project, regular meetings were held with company employees, where the state of the project at that time was discussed.

As a result, the application was recreated on a new platform. Also, new, more effective solutions for old tasks were developed: the user interface was updated, the application architecture was created, new functions were added. At the same time, all established requirements were met.

In conclusion, we can say that the application can be completely recreated on the new platform, and it will be better adapted to work on modern devices.

The thesis is in Estonian and contains 40 pages of text, 5 chapters, 35 figures, 0 tables.

## Lühendite ja mõistete sõnastik

UWP	<i>Universal Windows Platform</i>
WPF	<i>Windows Presentation Foundation</i>
MVVM	<i>Model-View-ViewModel</i>
XML	<i>Extensible Markup Language</i>
XAML	<i>Extensible Application Markup Language</i>
RQ	<i>Research Question</i>

## Sisukord

Autorideklaratsioon.....	2
Annotatsioon.....	3
Abstract Team project: Moving an existing application to UWP platform .....	4
Lühendite ja mõistete sõnastik .....	5
Jooniste loetelu .....	7
1 Sissejuhatus .....	9
1.1 Projekti lühikirjeldus .....	9
1.2 Projekti probleem .....	10
1.3 Projekti eesmärk.....	10
1.4 Teostatud funktsionaalsus.....	10
1.5 Töö edasine struktuur .....	12
2 Metoodika.....	14
2.1 Objekti detailne kirjeldus.....	14
2.2 Tööriistade kirjeldus.....	14
2.3 Tööprotsessi kirjeldus.....	15
3 Töö tulemused .....	17
3.1 Korralikult välja töötatud nõuded .....	17
3.2 Arhitektuur.....	18
3.3 Disain.....	20
3.4 Vaated.....	22
3.5 Kood .....	34
3.6 Testid .....	36
4 Analüüs ja järeldused .....	38
4.1 Osas 3 esitatud tehnilise teostuse põhjendus .....	38
4.2 Kirjanduse ülevaade .....	39
4.3 Teostatud tööde detailne kirjeldus logi vormis.....	43
4.4 Hinnang projekti teostamise protsessi kohta .....	47
4.5 Meeskondlik konsensuslik hinnang kõikide meeskonnaliikmete panuse kohta ...	48
5 Kokkuvõte .....	49
Kasutatud kirjandus .....	50
Lisa 1 – Enda panuse kirjeldus ja eneseanalüüs – Alexander Boyko.....	51
Lisa 2 – Enda panuse kirjeldus ja eneseanalüüs – Deniss Šutov.....	53

## Jooniste loetelu

Joonis 1. Rakenduse projektid. ....	18
Joonis 2. Andmebaasi mudel ja seosed. ....	19
Joonis 3. <i>MainPage</i> vaade. ....	22
Joonis 4. Heli lindistamise menüü. ....	23
Joonis 5. Raporti menüü. ....	23
Joonis 6. <i>AudioEditor</i> vaade(heli redigeerimine). ....	23
Joonis 7. <i>HomePage</i> vaade patsienti valemise tabeliga. ....	24
Joonis 8. <i>HomePage</i> vaade uuringu valimise tabeliga. ....	25
Joonis 9. <i>Image Editor</i> vaade(pildi redigeerimine). ....	26
Joonis 10. <i>Media</i> vaade(kõik uuringuga seotud failid). ....	27
Joonis 11. <i>Media</i> vaade filtri nupud, kui on aktiveeritud filter <i>In process</i> . ....	28
Joonis 12 <i>Media</i> vaade(arhiiv oleku ja lõpetamise kuupäeva ilmumine). ....	28
Joonis 13. <i>Patient`s report</i> vaade. ....	29
Joonis 14. <i>Pdf viewer</i> vaade(pdf-faili vaatamine rakenduses) ....	29
Joonis 15. <i>Print preview</i> vaade(printimise eelvaade) ....	30
Joonis 16. <i>Report</i> vaade ....	30
Joonis 17. <i>Settings</i> vaade(rakenduse seadised). ....	31
Joonis 18. <i>Media</i> vaade näides tume teemaga. ....	31
Joonis 19. <i>Templates</i> vaade(rakenduse mustrid) ....	32
Joonis 20. <i>Video editor</i> vaade(video redigeerimine). ....	33
Joonis 21. <i>Video viewer</i> vaade(video vaatamine rakenduses). ....	34
Joonis 22. Klassi <i>Database</i> meetodid andmebaasiga suhtlemiseks. ....	35
Joonis 23. Klassi <i>Shapes Factory</i> meetodid objektide joonistamiseks. ....	35
Joonis 24. Testide ülevaade. ....	37
Joonis 25. Projekti <i>commit</i> -ide arv. ....	43
Joonis 26. Projekti <i>commit</i> -ide arv. ....	43
Joonis 27. Projekti <i>commit</i> -ide arv. ....	43
Joonis 28. Käsitsi koostatud tegevuste logi. ....	44
Joonis 29. Käsitsi koostatud tegevuste logi. ....	44

Joonis 30. Käsitsi koostatud tegevuste logi.....	45
Joonis 31. Käsitsi koostatud tegevuste logi.....	45
Joonis 32. Käsitsi koostatud tegevuste logi.....	45
Joonis 33. Käsitsi koostatud tegevuste logi.....	46
Joonis 34. Käsitsi koostatud tegevuste logi.....	46
Joonis 35. Käsitsi koostatud tegevuste logi.....	46



# 1 Sissejuhatus

Tänapäeval tarkvaraarendus ei seisa paigal ja aktuaalsed lahendused võivad paar aastaga vananeda. Tänu suurele konkurentsile turul peab iga ettevõtte säilitama oma tarkvara kõrge kvaliteeti ning koguaeg arenguga kaasas käia. Vananeda aga saavad ka platvormid ja üleminek uuele platvormile mõnikord võtab väga palju aega ja ressursse.

Käesoleva projekti raames oli tehtud olemasoleva rakenduse analüüs. Kuna kõik nõuded oli ettevõtte poolt ette antud ja kõik meie poolt tehtud muudatused olid ettevõttega samuti läbiräägitud. Suuremad otsused mida me saime iseseisvalt teha on projekti arhitektuuri ja arendusmetoodika valimine.

Projekti analüüsimiseks oli püstitatud kolm uurimisküsimust. Esimene uurimisküsimus on “*UWP* eelised ja puudused”. Teine uurimisküsimus on “Kaasaegne meedia failide redigeerimise tarkvara”. Kolmas uurimisküsimus on “*MVVM* arhitektuuri eelised ja puudused”. Uurimisküsimuste vastuste saamiseks oli uuritud sarnase märksõnadega teadusartiklid.

## 1.1 Projekti lühikirjeldus

Olemasolev rakendus on suur rakendus haiglatele, mis vastavalt versioonidele omab erinev funktsionaalsus. Peamised rakenduse versioonid on *Pathology* – mida kasutavad haigla patoloogia osakonna töötajad. *Surgery* – mida kasutavad kirurgia osakonna töötajad ja *Editor* – mida kasutakse kogutud andmete kuvamiseks ja märkimiseks. Meie eesmärk oli olemasoleva rakenduse *Editor* versiooni taasloomine uuel platvormil. Sellel versioonil saavad haigla töötajad kogutud uuringu käigus meedia failid vaadata. Näiteks kehade pildid või kogu uuringu videosalvestus. Samuti selle versiooni abil saavad kasutajad teha märkmed. Näiteks piltide peal joonistada ja lõikamiskohad märkida või lindistada heli kommentaaridega. Olemasolev rakendus on kirjutatud *WPF* platvormil ja kuna me ei saanud taaskasutada olemasoleva koodi, siis kogu funktsionaalsus oli uuesti kirjutatud. Suurem osa nõudeid oli meile ette antud olemasolevast rakendusest, aga saime

ka täiendavad nõuded arendamise käigus. Ning oli ka ettevõtte poolt fikseeritud platvorm ja disain. Loodud rakendus on *Windows* operatsioonisüsteemile suunatud tarkvara, kus kasutajad (arstid) saavad patsientide andmeid vaadata, filtreerida, muutada ja kustutada. Samamoodi uuringu meedia failide (pildid, videod, heli) redigeerimine, heli salvestamine, piltide peal joonistamine, raportite tegemine, printimine, mustrite avamine (pildid, videod, pdf ja dokumendid).

## **1.2 Projekti probleem**

Tarkvara mis on tänapäeval kasutusel on tehnoloogiliselt vananenud ega vasta kasutajate ootustele. Kuna rakendus on juba kasutusel kaua aega, mõned kasutajaliidese disaini elemendid on juba vananenud ning vajavad uuendamist. Probleem on ettevõtte poolt formuleeritud ning saadud kasutajate kogemuste põhjal.

## **1.3 Projekti eesmärk**

Lõputöö peamiseks eesmärgiks on meditsiinilise meedia käsitlemise tarkvara rekonstrueerimine *Windows 10* platvormil. Ning pakkuda haigla personaalile, kaasaegset, intuitiivset (ehk sarnane olemasoleva rakendusega, et säilitada kasutaja kogemust) kasutust omavat tarkvara, mis oleks hallatav ja kuhu oleks lihtne uusi omadusi lisada. Samuti ka uurida uue platvormi ning selle eelised ja puudused.

## **1.4 Teostatud funktsionaalsus**

Suurem osa nõuetest oli saadud olemasolevast rakendusest ja mõned nõuded tekkisid ka projekti jooksul. Esialgne plaan oli taaskasutada kirjutatud koodi ja teisendada seda uuele platvormile, uuendatud kasutajaliidese disainiga. Kuid analüüsi käigus selgus, et olemasolev koodi ei ole võimalik taaskasutada ja pidime kogu koodi uuesti kirjutama.

Projekti käigus oli teostatud:

### **Funktsionaalsed nõuded**

- Patsiendi ja uuringu valimine, filtreerimine (põhiandmete vaatamine).

- Meedia failide (pildid, heli, video) vaatamine, filtreerimine (meedia tüübi järgi, staatuse järgi), otsimine, lisamine, kustutamine.
- Valitud meedia failide redigeerimine (joonistamine, lõikamine, teksti lisamine, videost pildi tegemine).
- Pildi kalibreerimine, et näidata joonistatud objektide reaalsed mõõdud.
- Konteineri ja kassettide lisamine ja kasutamine pildi redigeerimisel.
- Dokumentide (*Patient's Report, Report, Preview*) loomine valitud meedia failidega.
- Heli lindistamine ja salvestamine seda uue meedia failina.
- Mustrite lisamine ja redigeerimine (pildid, videoid, dokumendid, pdf-failid).
- Meedia failide staatuse muutmine (lõpetatud, lõpetamata, jagatud).
- Arhiivi saatmise staatuse näitamine.

### **Mittefunktsionaalsed nõuded**

- Kiire ühendus andmebaasiga, kui töötate suure hulga patsientidega.
- Originaal meedia failid peavad visuaalselt eristuma redigeeritud failidest.
- Kasutaja saab navigeerimismenüü abil igal hetkel avada vajalik vaade.
- Arendus platvormiks tuleb kasutada *UWP*.
- Arhitektuuriks tuleb kasutada *MVVM*.
- Kasutajaliides pole detailidega ülekoormatud ning on sarnane originaal-rakendusega.
- Rakendus peab kasutama olemasoleva andmebaasi.
- Uuringu valimisel kuvatakse ainult selle uuringuga seotud andmed.

- Redigeeritud faile saab korduvalt redigeerida (elementide kustutamine, lisamine).
- Lokalisatsioon ehk keelte valik ja ka lihtne võimalus keeli juurde lisada.
- Joonistatud objektide salvestamine *XML* faili, et saaks neid muuta või kustutada hiljem.

### **Kasutusjuhtude kirjeldus**

- Patsiendi/uuringu valimine: Esilehel valida patsiendi -> valida tema uuring -> avaneb lehekülg, kus on olemas kõik meedia failid valitud uuringu kohta.
- Meedia faili redigeerimine: Meedia-lehel topelt vajutamine meedia faili peale -> avaneb lehekülg, kus on võimalik redigeerida valitud meedia faili (vastavalt faili tüübile avaneb sobiv lehekülg).
- Pildi peal joonistamine: Pildi redigeerimis lehel tuleb valida joonistamis instrumendi -> valida instrumendi parameetrid (värv, paksus) -> joonistada pildi peal -> salvestada all oleva nupuga.
- Dokumenti koostamine: Meedia vaatel valige failid, mida te tahate lisada dokumendile -> vajuta *Patient's Report/Report/Print Preview* nuppu (sõltuvalt sellest, millist dokumenti soovite koostada).
- Settingute muutmine: Settings vaatel valida sobivad parameetrid -> vajuta "Save Changes" nuppu.
- Heli salvestamine: Vajuta *Audio Record* nuppu -> avanenud aknal vajuta *Record* selleks et alustada helisalvestus -> Vajuta *Stop*, kui sooviks lõpetada -> Vajuta *Play*, kui sooviks lindistatud heli kuulata -> Vajuta *Save*, kui tahate salvestada lindistatud heli.

## **1.5 Töö edasine struktuur**

Käesolev lõputöö koosneb viiest peatükist. Esimeses nendest on toodud ülevaade kasutatud tehnoloogiatest ja tööprotsessist ning ülevaade asjadest, mis oli juba olemas enne projekti alustamist. Teises peatükis kirjeldatud tehtud tööde tulemused ja tehniline

dokumentatsioon. Kolmas peatükkis on tehtud analüüs ja järeldused ning töö tulemuste põhjendus, valideerimine ja hinnang projekti teostamise protsessi kohta. Neljandas peatükkis kirjutatud projekti kokkuvõtte ja viiendas on isiklik aruanne iga meeskonnaliige kohta.

## 2 Metoodika

### 2.1 Objekti detailne kirjeldus

Rakenduse kasutajad on haigla töötajad, kelle peamine eesmärk on uuringuga seotud meedia failide vaatamine ja redigeerimine. Kõik failid asuvad serveris, täiendav informatsioon failide kohta aga asub andmebaasis, sealhulgas ka teekond kaustale kus failid paiknevad. Rakenduse abil saab kasutaja vaadata andmebaasis olevad andmed koos serveris oleva meedia failidega. Sealhulgas on patsiendid, uuringud, arstid, meedia failid, uuringu tüübid, uuringuga seotud konteinerid ja kassetid.

Pärast uuringu valimist saab sellele lisada uued meedia failid. Näiteks pildid, videod või lindistada heli. Piltidest saab teha raporteid koos uuringu informatsiooniga ja printida need raporteid. On olemas võimalus piltide peal joonistada ja need muudatused salvestada. Kuna originaalne pilt peaks jääma nii nagu ta on, siis kõik joonistatud objektid salvestatakse eraldi failisse ja pildi avanemises joonistab rakendus kõik objektid uuesti originaal pildi peal. Videotest ja heli failidest saab vajalikud osad lõigata ning lisaks videost saab ka eraldi pildi teha ja salvestada seda uuringu meedia failina. Igal rakendusel lokaalses võrgus on võimalus vaadata mustreid ehk *Templates*. Need saavad olla kas pildid, videod, dokumendid või pdf-failid.

Rakenduse on võimalik käivitada mitmes keeles(praegu töötavad inglise ja eesti keeled). Samuti on väga lihtne uusi keeli lisada.

### 2.2 Tööriistade kirjeldus

*Universal Windows Platform* (edaspidi *UWP*) rakendus on kirjutatud C# programmeerimiskeeles ja tööristjana kasutatud *Microsoft Visual Studio 2019*. Andmebaasi haldus süsteemiks oli kasutatud *Microsoft SQL Server*. Projekti raamistikuks oli kasutatud *Model-View-ViewModel* (edaspidi *MVVM*). Koodi refaktoormiseks oli kasutatud *ReSharper Ultimate*. Kasutajaliidese elementide defineerimiseks oli kasutatud

*Extensible Application Markup Language* (edaspidi *XAML*). Rakenduse versioonihalduseks oli *GitLab*.

## 2.3 Tööprotsessi kirjeldus

Projekti alguses oli tehtud olemasoleva rakenduse analüüs. Mille käigus selgus, et olemasolev rakendus on Windows Presentation Foundation (edaspidi WPF) rakendus ja suurem osa koodi me ei saanud taaskasutada kuna meie rakendus kasutab teist raamistiku (.NET Framework WPF-i puhul ja .NET Standard UWP puhul). .NET Framework kasutab mõned NuGet pakettid, mis puuduvad .NET Standard-i raamistikus ja sellepärast ei oli võimalik taaskasutada kogu olemasoleva koodi. Kuna pidime olemasolevast rakendusest ainult väikese osakese taasluua, siis otsustasime kogu koodi uuesti kirjutada ja taasluua originaalse rakenduse funktsionaalsus uuel platvormil nullist. Abiks oli olemasolev rakendus, mille koodi me kasutada küll ei saanud, aga kasutajaliidese disain(nuppude ja andmete asukohad) ei olnud nullist leiutatud vaid taaskasutatud.

Kuna meie plaanid ja ülesanded ei olnud planeerimisetapis väga täpselt defineeritud siis otsustasime kasutada agiilset arendusmetoodikat. See sobiks meie väga hästi, sest meie meeskonnas on ainult kaks liiget ja arenduse käigu jagamine sprintideks ja sagedased koosolekud aitasid meile säilitada distsipliini ja ülevaade meeskonna tulemustest. Meie sprinti kestus oli orienteeruvalt üks kuu ja igaüks tegi keskmiselt 20 tundi tööd nädalas. Omavahelised koosolekud toimusid vähemalt kaks korda sprindi jooksul, kus me jagasime oma mõtted, tulemused ja probleemid.

Sprinti lõpus toimus koosolek kus me näitasime oma vahetulemused ettevõttele. Seal näitasime tehtud töö ja arutlesime uued ülesanded ja nõuded. Sellise koosoleku lõpus oli meil jälle olemas ülesanded uueks sprindiks.

Kõik ülesanded, nii tehtud kui ka tegemata, oli kogu aeg igaühele kättesaadavad *Google Disk*-is. Igaüks tegeles sellise ülesandega, mida ta oli võtnud. Kui aga selgus, et ülesanne on liiga keeruline, proovisime seda jagada pooleks. Et ülesannete keerukus oleks igal ülesandel samaväärne.

Esimesed sammud olid tehtud koos selleks, et süveneda ja aru saada suure pilti. Kuna nii platvorm, kui ka disain oli teatud siis hakkasime arutlema rakenduse arhitektuuri.

Arutluse käigus oli otsustatud kasutada mitmekihilise rakenduse arhitektuuri, mida me juba kasutasime ülikoolis tehtud projektide realiseerimiseks. Samuti defineerisime koodi kirjutamise reeglid. Kasutasime ka paarisprogrammeerimis metoodikat, kus üks meeskonnaliige kirjutas koodi ja teine liige vaatas ja kontrollis samal ajal. Selline metoodika aitas meile õppida koos ja kohe arusaamatused läbi arutada.

Oli vaja väga palju ka juurde õppida ja uurida, sest ei ole keegi liikmetest *UWP* rakendustega ennem kokku puutunud. See võttis alguses kõige suurema ajakulu. Selleks oli loetud *Microsoft*-i poolt kirjutatud dokumentatsioon *UWP* platvormi kohta. Igaüks jagas olulised asjad ja informatsioon selle osa kohta, mida kindlasti töö käigus oleks vajalik kõikidele meeskonnaliikmetele.

Kuna meeskonnas on ainult kaks liiget, siis proovisime kogu aeg jagada oma mõtteid ja ideid, mida saab paremini teha või mida tuleb juurde tegema. Selline tagasiside aitas vaadata probleemile või eesmärgile teisest nurgast. Kuigi alguses programmeerisime palju kahekesi, siis juba pärast esimest sprinti hakkasime ka tööd jaotama.

Töö oli jaotatud vastavalt ülesannete nimekirjale. Aga võib öelda, et esimese liige peamine ülesanne oli seotud piltide redigeerimisega, konteineritega, kassettidega ning koodi refaktoorisega. Ja teine liige tegeles kasutajaliidese disainiga, andmebaasi ühendusega, lokaliseerimisega, meedia failide andmete kuvamisega ning heli ja video redigeerimisega.



## 3 Töö tulemused

### 3.1 Korralikult välja töötatud nõuded

Enne projekti alustamist ettevõtte poolt oli antud järgmised mittefunktsionaalsed nõuded:

- Kasutada realiseerimiseks *UWP* platvormi.
- Kasutada disainiks *MVVM* arhitektuuri.
- Kasutada olemasoleva andmebaasi.
- Kasutajaliides pole detailidega ülekoormatud ning on sarnane originaaliga.

Ja kui meile tutvustati olemasoleva rakenduse, siis anti ka uue rakenduse peamised funktsionaalsed nõuded. Selle hulgas on:

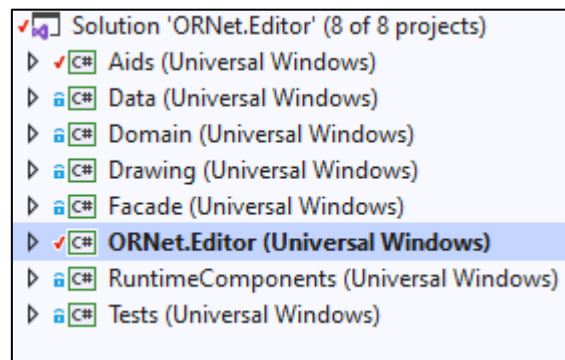
- Patsiendi ja uuringu andmete kuvamine, muutmine, filtreerimine, otsimine.
- Uuringu piltide, video- ja helisalvestuste järjestamine ja filtreerimine.
- Piltidele joonistamine(joon, nool, ring, nelinurk ja tekst) ja mõõdistused ning tehtud muudatused salvestada.
- Video- ja helisalvestuste lõikamine ja liitmine.
- Meedia failide kuvamine, filtreerimine, otsimine, oleku muutmine (lõpetatud/lõpetamata), lisamine, vaatamine esitlusvaates, kustutamine.
- Meediafailide eksportimine.
- Uuringu meedia failidest koostada ja printida dokumenti.
- Võimalus heli lindistada ja salvestada.

Projekti jooksul lisaksid veel mõned nõuded, nii funktsionaalsed kui ka mittefunktsionaalsed. Näiteks:

- Mustreid lisamine, filtreerimine, otsimine, vaatamine.
- Võimalus videost pilti teha ja selle pildi salvestada.
- Võimalus kasutada rakenduse erinevates keeltes.
- Võimalus konteinerite ja kassettide informatsioon peita.
- Võimalus rakenduse abil konsultatsiooni teha(video ja audio side).

### 3.2 Arhitektuur

Projekti arendamisel oli kasutatud mitmekihiline arhitektuur kus baasloogika(*Aids*, *Drawing*), andmekiht(*Data*), äri loogika(*Domain*), esituskiht(*Facade*, *ORNet.Editor*) ja testid(*Tests*) moodustavad kokku viis kihti. Kogu rakendus on jagatud kaheksaks projektideks ja näeb välja järgmiselt:



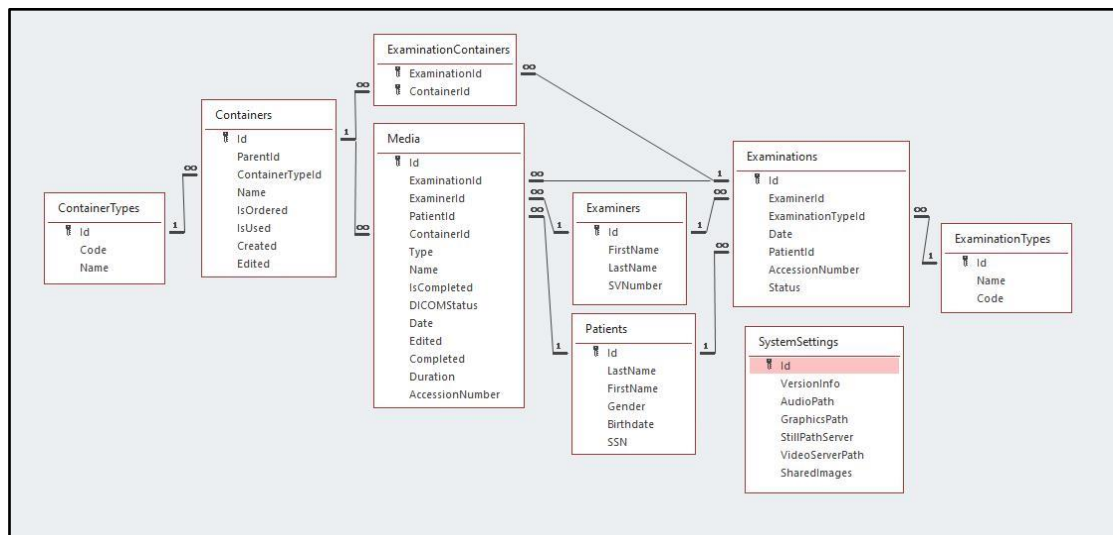
Joonis 1. Rakenduse projektid.

- ***Aids*** – projekt, mis sisaldab abiklassid ja elementaarloogika, mida tihti kasutatakse. Sealhulgas on konverterid, printimise ja heli lindistamise abiklassid, “string”-i generaatorid.
- ***Data*** – projekt, mis sisaldab andmete käsitlemise klassid. Näiteks andmebaasiga suhtlemine ja “xml” failide lugemine ja kirjutamine toimub selle projekti järgi.
- ***Domain*** – projekt, mis sisaldab andmebaaside mudelid ja XML objektide mudelid.

- **Drawing** – projekt, kus on kogu joonistamisega ja piltide redigeerimisega seotud loogika.
- **Facade** – projekt, mis sisaldab vaate mudelid, mudelite vabrikud(factory) ja vaadete parameetrid.
- **ORNet.Editor** – põhiprojekt, mis sisaldab rakenduse vaated, vaadete komponendid(user controls), pildid, logod ja lokaliseerimisega seotud failid.
- **Runtime Components** – projekt, mis sisaldab ainult ühe klassi ja võtab videost pildi(snapshot).
- **Tests** – projekt, mis sisaldab ülaltoodud projektide testid.

## Andmebaasi arhitektuur

Nagu oli juba mainitud me taaskasutame olemasoleva andmebaasi. Alloleval pildil on näha tabelid, mida rakendus kasutab. Aga olemasoleva rakenduse andmebaas on suurem, pildil on ainult need tabelid, mida kasutab meie taasloodud rakendus.



Joonis 2. Andmebaasi mudel ja seosed.

Kõige tihedamini kasutatakse tabelid *Patients* ja *Media*, mis hoiavad patsiendi ja meedia objekti informatsiooni. Joonisel 2 on näha tabelite seosed, mille järgi on üks mitmele seos patsiendi ja uuringu(*Examination*) vahel ja patsiendi ja meedia vahel. Mis tähendab, et igal uuringul ja meedia failil saab olla ainult üks patsient, aga patsiendil saab olla mitu uuringuid ja meedia faile. Uuringul samamoodi saab olla mitu meedia faile, aga uuringu

tüüp ja arst on uuringul alati üks. Tabel *ExaminationContainers* realiseerib mitu-mitmele seos uuringu ja konteineri tabelite vahel. Ehk igal uuringul saab olla mitu konteineri ja konteineril saab olla mitu uuringud.

Tabel *Media* hoiab vajalik informatsioon meedia faili kohta. Seal hulgas on millele uuringule, patsiendile see kuulub ja millise konteineriga see on seotud. Samamoodi ka selle faili tüüp, staatused ja vajalikud kuupäevad. Tabelis *SystemSettings* on rakenduse seaded ja muu informatsioon. Näiteks rakenduse versioon ja teekond kaustale, kus asuvad meedia failid.

### **Oluline märkus konteineri ja kasettide kohta:**

Tavaliselt meedia pilt on elundi pilt ja haiglas seda lõigatakse osadeks. Iga osa hoitakse eraldi kassetis. Ja kõik kassetid asuvad konteineris, järeltõlt igal meedia failil saab olla ainult üks konteiner(mille id hoitakse tabelis *Media ContainerId*-na). Andmebaasis konteinerid ja kassetid on esitatud ühe tabelina *Containers*. Neid saab eristada *ContainerTypeId* järgi ja *ParentId* seob millisele konteinerile kuulub kasset. Konteineri juhul on *ParentId* võrdne selle konteineri *Id*-ga.

## **3.3 Disain**

### ***Model-View-ViewModel***

Rakenduse arendamisel oli kasutatud *MVVM* disain, mis oli ettevõtte poolt etteantud. Kus *Model* on andmebaasist saadud andmed, mis asuvad projektis *Domain*.

*ViewModel*-id on andmebaasi mudelist tehtud klass, mis omab vajaliku informatsiooni andmete kuvamiseks. Kõik *ViewModel*-id asuvad projektis *Facade* ja nende loomiseks on olemas tehased ehk *Factory*, mis võtab parameetriks mudelite nimekiri ja tagastab vaate-mudeli nimekiri.

Kõik vaated asuvad põhiprojektis *ORNet.Editor* kaustal *Views*.

### **Piltide haldamine**

Nagu oli juba mainitud kõik failid asuvad serveris ja andmebaasist on kättesaadav ainult teekond ja failiga seotud informatsioon. Iga pildiga on aga seotud neli faili: pilt ise,

kompositsioon, pildi mall ja *XML* fail. Igal pildiga seotud failil on oma roll. Originaal pilt peab alati jääma muudatusteta ja *XML* failis on rakendusele vajalik informatsioon joonistatud objektide kohta, mis annab võimaluse neid pildi avamisel taas joonistada ja ka kustutada kui see on vajalik. Kompositsioon on originaal pilt koos joonistatud objektidega, mida kasutatakse raporti tegemiseks. Ja mall on väiksema resolutsiooniga originaal pilt koos joonistatud objektidega, mida kasutatakse kiireks kuvamiseks meedia nimekirjas.

### **Kasutajaliidese disain**

Kasutajaliidese realiseerimiseks jälgisime minimalistliku printsiibi. Sest rakenduse kasutavad haigla töötajad tööajal mis tähendab, et kasutajaliides ei pea olema tähelepanu kõrvalejuhtimise faktoritega ülekoormatud.

Kõik nupud ja kasutajaliidese elemendid, millega kasutaja regulaarselt kokku puutub, peavad olema suured. Et ei tunne kasutaja väsimust selleks, et on vaja kogu aeg otsida väikesed elemendid ekraanil.

Olemasolev rakendus oli meie jaoks näidis, mida on juba kaua aega kasutusel ja mille disain on juba tuttav. Seega otsustasime säilitada põhiliste nupud asukohad.

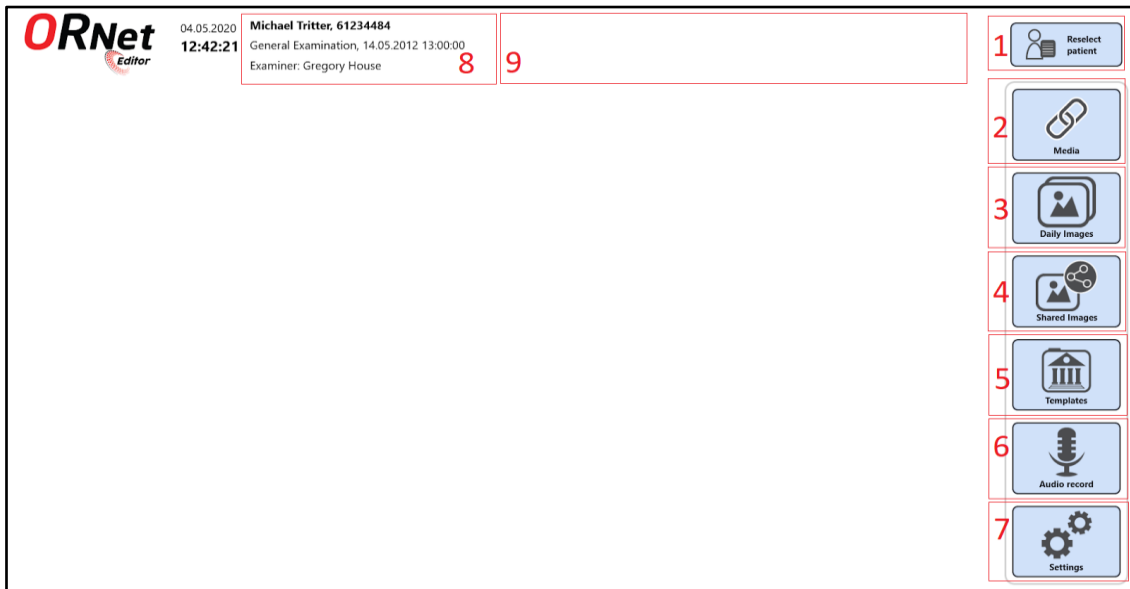
Peamised muudatused olid tehtud elementidega, mis ei ole rohkem kasutusel, neid kustutasime ära. Kõik muudatused mida me olime teinud oli kokkulepitud ja näidatud ettevõttele. Kui ei olnud nad tulemustega rahul, siis oli see muudetud vastavalt nende ootustele. Näiteks selleks olemasolevas rakenduses pärast uuringu valimisest, ilmub kõik uuringud meedia vaatel, mis kindlasti võtab rohkem aega ja ressursse andmete laadimiseks. Meie lahendus oli selline, et pärast uuringu valimist ilmub ainult selle uuringuga seotud andmed ja uuringu vahel saab väga mugavalt liikuda üleval oleva nupuga(vaata Joonis 3 kirjeldus number 8). Samuti vaatel *Image Editor* on teisendatud joonistamise tööriistad vasakult poolt ülevale, mis annab võimaluse pildile rohkem kohta jääda.

Kõik ikoonid ja logod on päritud olemasoleva rakendusest. Kuna *UWP* platvorm omab oma unikaalsed märgid, siis kasutajaliideses kasutatakse see võimalus. Kuigi me kasutasime uued elemendid, ikka jääb kasutajaliides tuttavaks kasutajatele, mis aitab nendele kiiresti orienteeruda uuel rakendusel.

### 3.4 Vaated

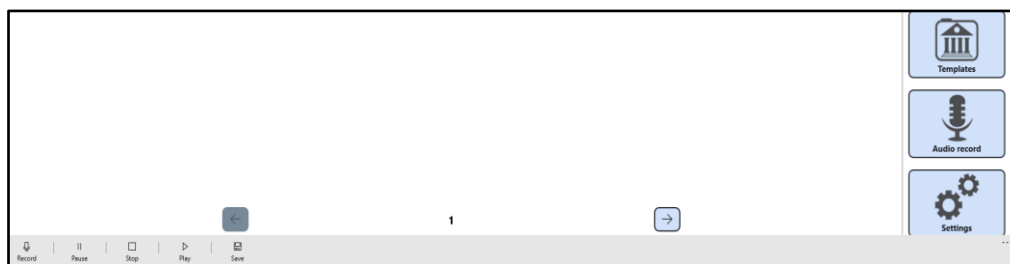
#### *Main page*

Peamine vaade on *MainPage* mis koosneb navigeerimis nuppudest ja valitud patsiendi ja uuringu informatsioonidest. Keskele ilmub valitud vaade.



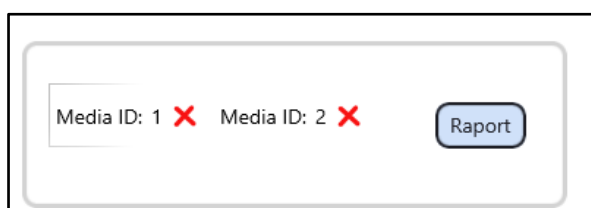
Joonis 3. *MainPage* vaade.

1. *Reselect Patient* – avaneb vaade *HomePage* kus saab uuesti patsiendi valida.
2. *Media* – avaneb vaade *Media* kus on kõik uuringuga seotud failid.
3. *Daily Images* – sarnaselt *Media* vaatega, aga failid on filtreeritud tänase kuupäeva järgi.
4. *Shared Images* – sarnaselt *Media* vaatega, aga failid on näidatakse ainult jagatud olekuga pildid.
5. *Templates* – avaneb vaade *Templates*, kus on kõik rakendusele lisatud mustrid.
6. *Audio Record* – avaneb heli lindistamise menüü.



Joonis 4. Heli lindistamise menüü.

7. *Settings* – avaneb vaade *Settings*, kus saab rakenduse seaded muuta.
8. *Patient Info* – patsiendiga ja uuringuga seotud informatsioon, uuringu peale klikkides avaneb “*HomePage*” vaade, kus saab uuesti uuringu valida.

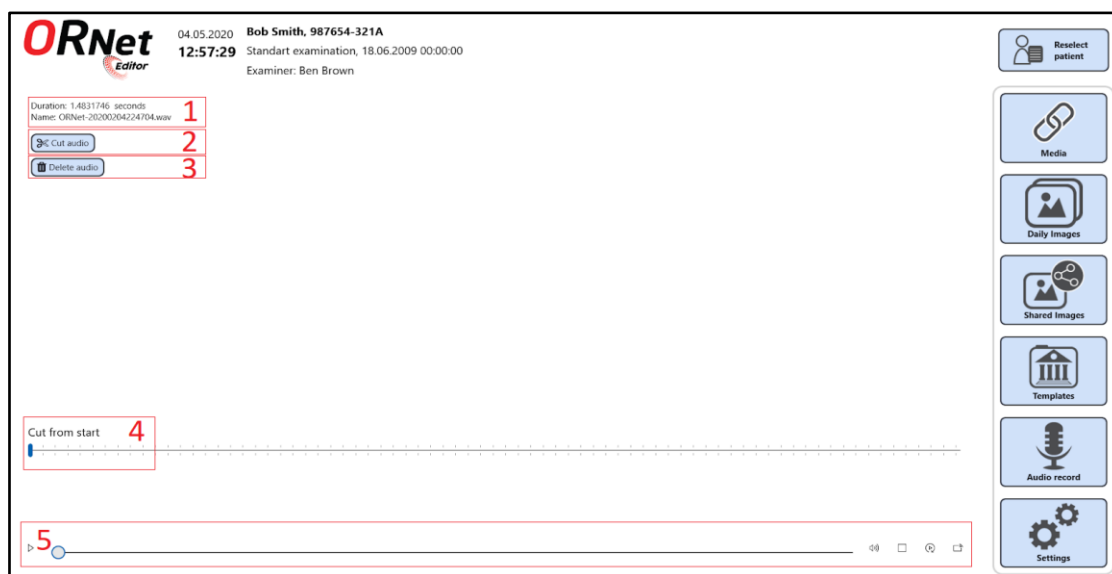


Joonis 5. Reporti menüü.

9. *ReportId List* – erinevatest uuringutest saab *Media* vaatest lohistada pildid raporti menüüsse. Vajutades *Raport* nupule avaneb *Report* vaade piltidega erinevatest uuringutest.

### Audio editor

Audio faili redigeerimise vaade. Siin saab audio faili kuulata, lõigata ja kustutada.

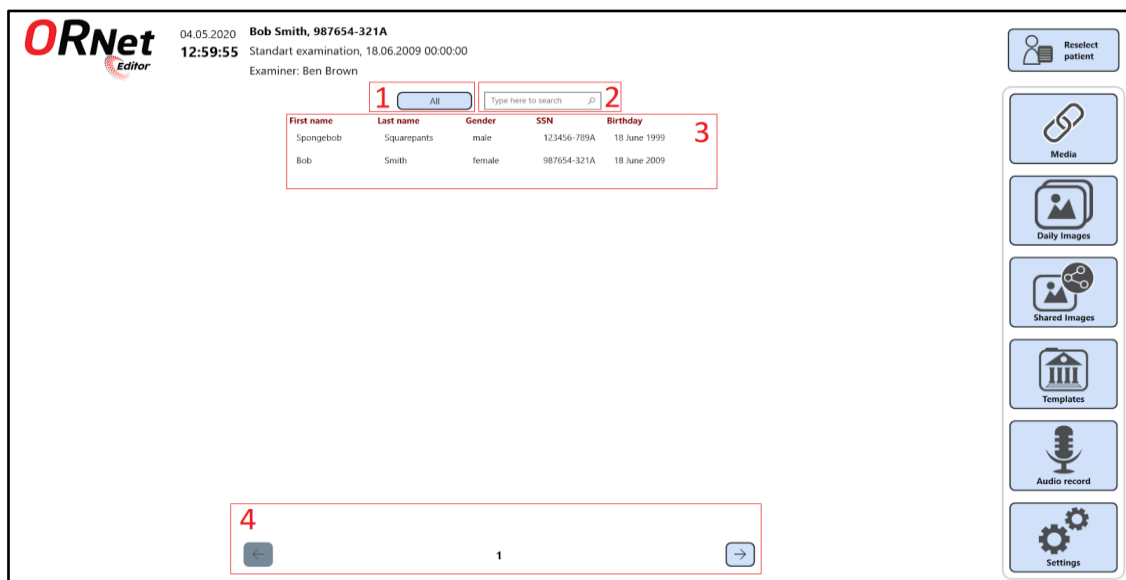


Joonis 6. AudioEditor vaade(heli redigeerimine).

1. *Audio info* – heli faili informatsioon(faili nimi ja kestus).
2. *Cut Audio* – nupp heli faili lõikamiseks algusest(fail lõigatakse algusest lõpuni ehk lõigata saab ainult algusest).
3. *Delete Audio* – nupp heli faili kustutamiseks.
4. *Cut from start* – seadis, mille abil saab lõikemise algus.
5. *Player* – audio pleier.

### Home Page

Vaade, mis ilmub rakenduse käivitamisel ja näitab kõik andmebaasis olevad patsiendid. Patsiente saab otsida ja filtreerida ning valida.



Joonis 7. HomePage vaade patsienti valemise tabeliga.

Pärast patsiendi valimist ilmub tabel valitud patsiendi uuringutega. Kui patsiendil on ainult üks uuring see valitakse automaatselt. Patsiendi ja uuringu valimine tähendab, et kõik kuvatud andmed on seotud konkreetse patsiendiga ja konkreetse uuringuga. Samuti kõik lisatud failid ja lindistatud heli salvestatakse seostatakse valitud uuringuga ja patsiendiga.



04.05.2020 Michael Tritter, 61234484  
 13:01:32 Standart examination, 18.06.2009 00:00:00  
 Examiner: Ben Brown

Examination Type	Date	Accession Number
General Examination	14.05.2012 13:00:00	3456
Premium examination	18.06.2009 00:00:00	3456

5

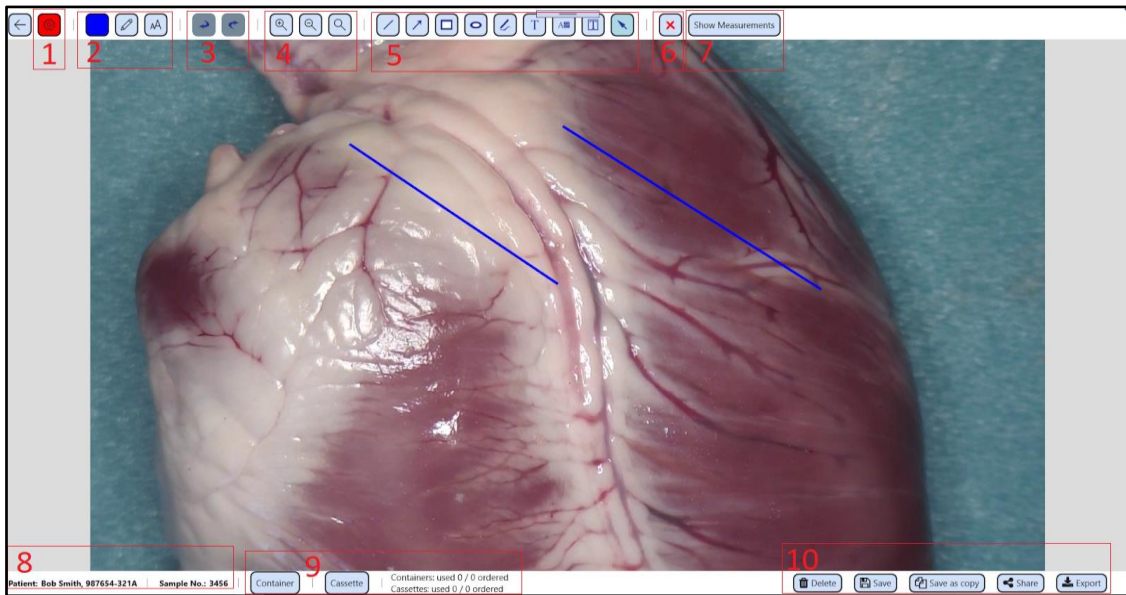
Reselect patient  
 Media  
 Daily Images  
 Shared Images  
 Templates  
 Audio record  
 Settings

Joonis 8. *HomePage* vaade uuringu valimise tabeliga.

1. *Patient Filter* – filter, mille abil saab patsiendid filtreerida kuupäeva järgi (filtri variandid on kõik, tänased, eelmise nädala).
2. *Patient Search* – patsientide otsing eesnimi, perekonnanimi ja isikukoodi järgi.
3. *Patient's list* – patsientide nimekiri.
4. *Pagination* – kui patsiente on rohkem kui seadistatud ühele leheküljele siis ilmuvad lehekülje number ja navigeerimis nupud (antud juhul on seadistatud 2 patsienti ühele leheküljele).
5. *Examination's list* – pärast patsiendi valimist ilmub uuringu valimis tabel. Kui uuring on ainult üks, siis rakendus valib seda automaatselt.

### ***Image editor***

Vaade võimaldab valitud pildi peal joonistada ning tehtud muudatused salvestada või salvestada koopiana. Samuti siin saab pildi kustutada, jagada ja eksportida (salvestada lokaalsel kettal valitud kaustal).



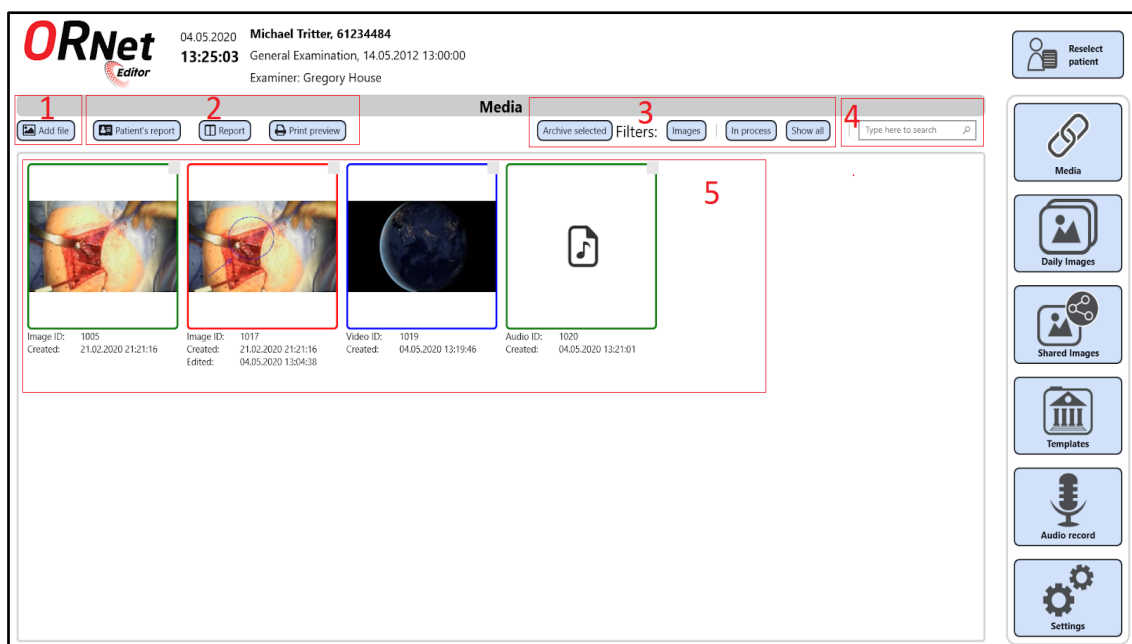
Joonis 9. Image Editor vaade(pildi redigeerimine).

1. *Calibration* – nupp pildi kalibreerimiseks, mille abil saab seadistada pikslite suhe millimeetritesse, selle suhe abil arvutatakse kõik mõõdud.
2. *Brush settings (color, thickness/text size)* – harja seaded(värv, paksus) ja teksti suurus.
3. *Undo/Redo* – nupud joonistatud objektide kustutamiseks ja taastamiseks.
4. *Zoom* – nupud pildi suurendamiseks.
5. *Tools* – nupud joonistamis objekti valimiseks(joon, nool, ring, nelinurk). Sealhulgas ka teksti lisamise nupp ja tihti kasutatud tekstide automaatne lisamine.
6. *Delete object* – nupp valitud objekti kustutamiseks.
7. *Show/Hide measurements* – nupp objekti mõõdete näitamiseks(pindala, maht, pikkus, perimeeter).
8. *Image information* – andmed meedia faili kohta.
9. *Containers* – nupud konteinerite ja kassettide valimiseks ja täiendav informatsioon pildiga seotud konteineritega ja kassettidega.

10. *Actions with picture* – nupud pildi kustutamiseks, salvestamiseks, salvestamiseks koopiana, jagamiseks(luuakse uus pilt olekuga jagatud), eksportimiseks(salvestamine lokaalsele kettale).

## Meedia

Vaade kuvab kõik uuringuga seotud meedia failid. Siin näidetakse failidega seotud informatsioon ning saab neid failid filtreerida ja nende seas otsida. Samuti saab meedia failid valida(paremal üleval nurgal ilmub linnuke kui fail on valitud) ja valitud failidest teha raporteid(*Report, Patient's report, Print preview*).



Joonis 10. *Media* vaade(kõik uuringuga seotud failid).

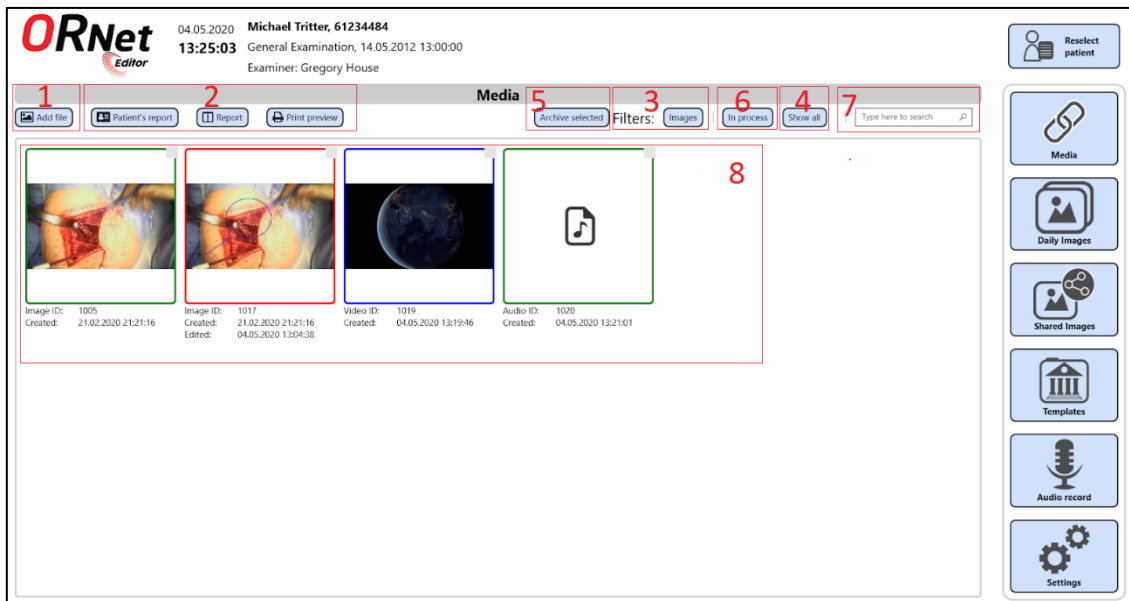
1. *Add File* – nupp uue faili lisamiseks(lisada saab ainult pildid ja videod).
2. *Documents* – valitud meedia failidest koostatakse raport.
3. *Filters* – filtrid tüübi(pilt, video, heli) või oleku(lõpetatud, lõpetamata) järgi.
4. *Show all* – tühistab kõik filtrid.
5. *Archive selected* – arhiveerib valitud failid ja näitab arhiveerimis oleku.
6. Kui on valitud filter lõpetamata, mis näitab lõpetamata failid. Siis ilmub *Complete* nupp, millega saab faili oleku muuta lõpetatuks.Kui on valitud filter lõpetatud siis ilmub nupp *Bring back*, millega saab oleku muuta tagasi lõpetamatuks.

7. *Search* – otsing meedia *ID* järgi.



Joonis 11. *Media* vaade filtri nupud, kui on aktiveeritud filter *In process*.

8. *Media files* – meedia failide nimekiri kus originaalsel pildil on roheline piir, redigeeritud pildil on punane piir ja video failidel on sinine piir. Pildi all on olemas informatsioon faili kohta(id, loomise kuupäev, redigeerimis kuupäev ja lõpetamise kuupäev juhul, kui objekti on lõpetatud olekuga)



Joonis 12 *Media* vaade(arhiiv oleku ja lõpetamise kuupäeva ilmumine).

### *Patient's report*

Vaade, millele saab lisada uuringu pildid ja mustrid kasutades *Add Template* nuppu. Automaatselt täidetakse lahtrid seotud uuringu informatsiooniga ning lisaks on kasutajal võimalus kirjutada kommentaari.

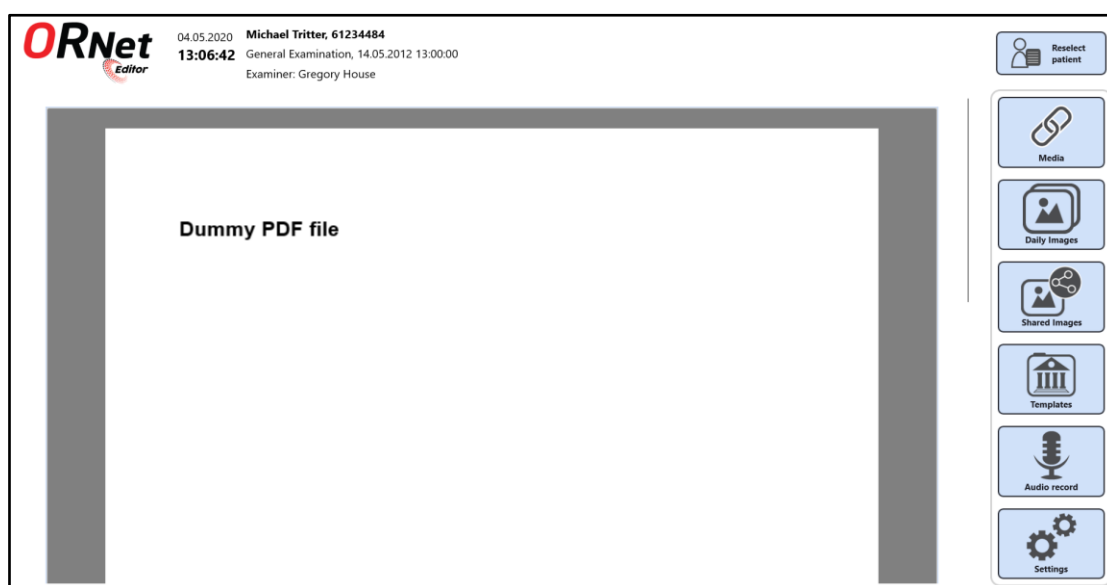


Joonis 13. Patient's report vaade.

1. *Print* – nupp mis avaneb Windows printimise eelvaade.
2. *Add Template* – nupp millega saab muustritest lisada pilt raportile.
3. *Remove Image* – pildi kustutamise nupp raportist.
4. *Basic Info* – andmed uuringu kohta.
5. *Doctor's comments* – raporti kommentari lisamise koht.

### *Pdf viewer*

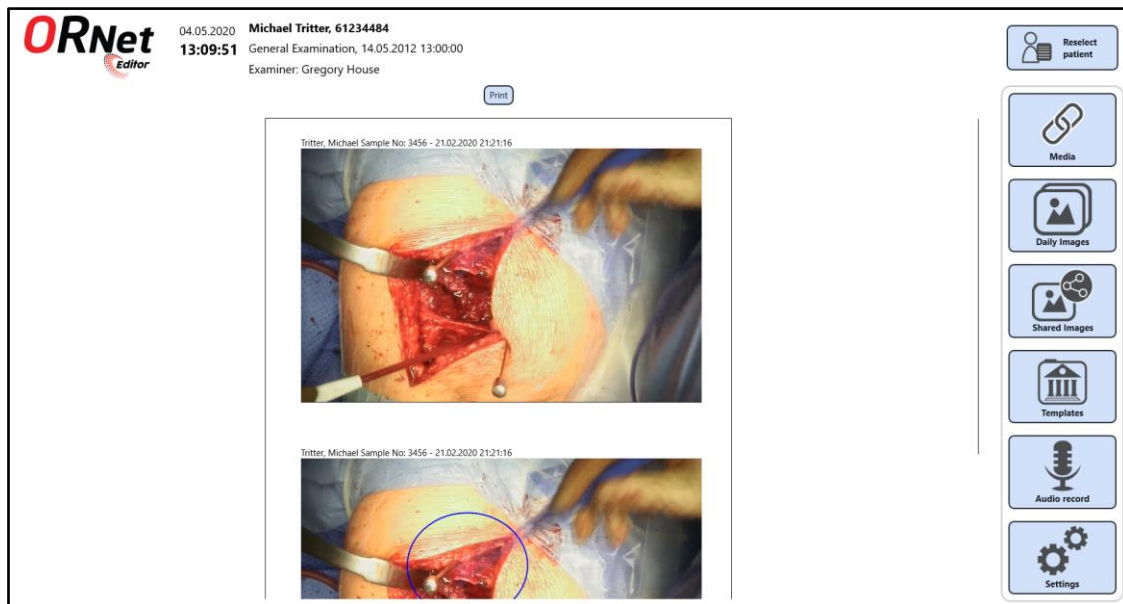
Pdf faili lehekülgede kuvamine rakenduses.



Joonis 14. Pdf viewer vaade(pdf-faili vaatamine rakenduses)

## Print preview

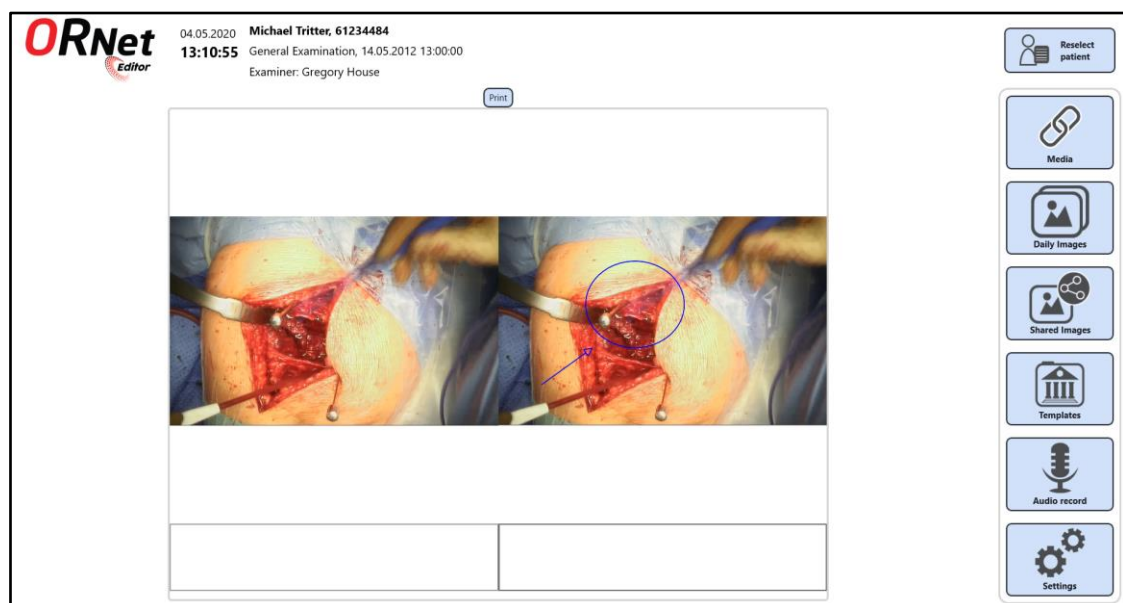
Printimise eelvaade kus ühele leheküljele ilmub pilt ja pildiga seotud patsient ja tegemise kuupäev.



Joonis 15. *Print preview* vaade(printimise eelvaade)

## Report

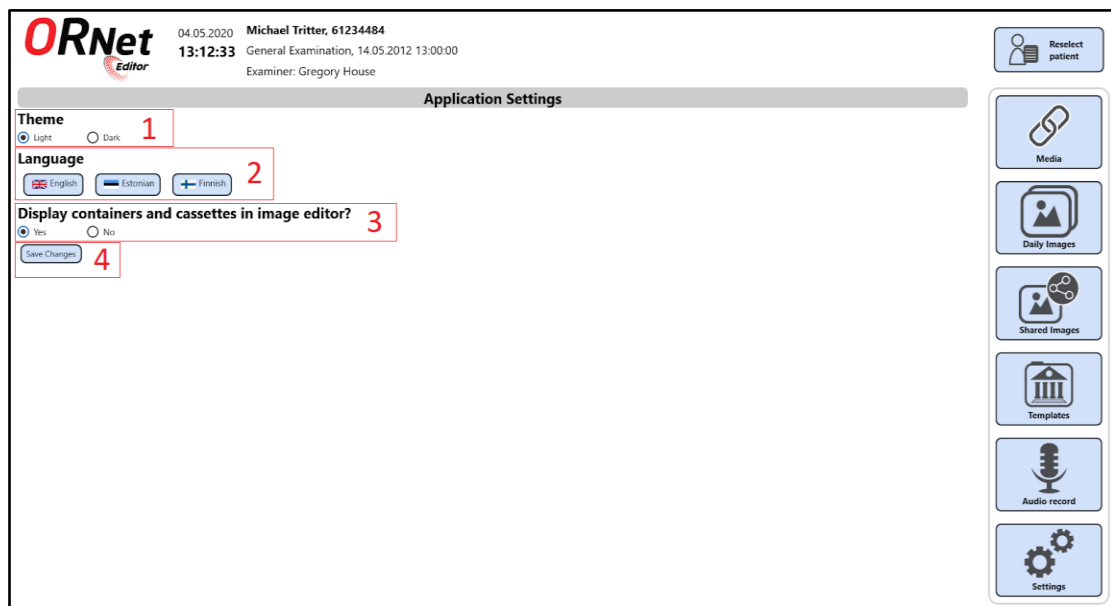
Vaade valitud piltidega kuhu saab käsitsi kommentaarid kirjutada ja tehtud aruanne ka printida.



Joonis 16. *Report* vaade

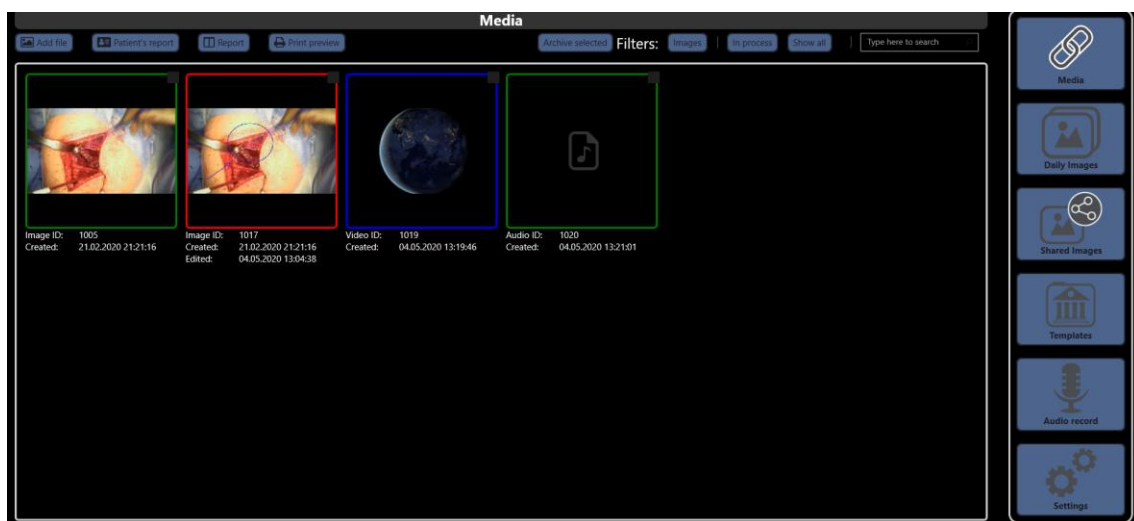
## Settings

Rakenduse seaded kus saab valida rakenduse teemat ja keelt. Ja vastavalt vajadustele näidata või peita konteinerid pildi redigeerimis vaatel.



Joonis 17. *Settings* vaade(rakenduse seadised)

1. *Theme* – rakenduse teema valimise nuppud.

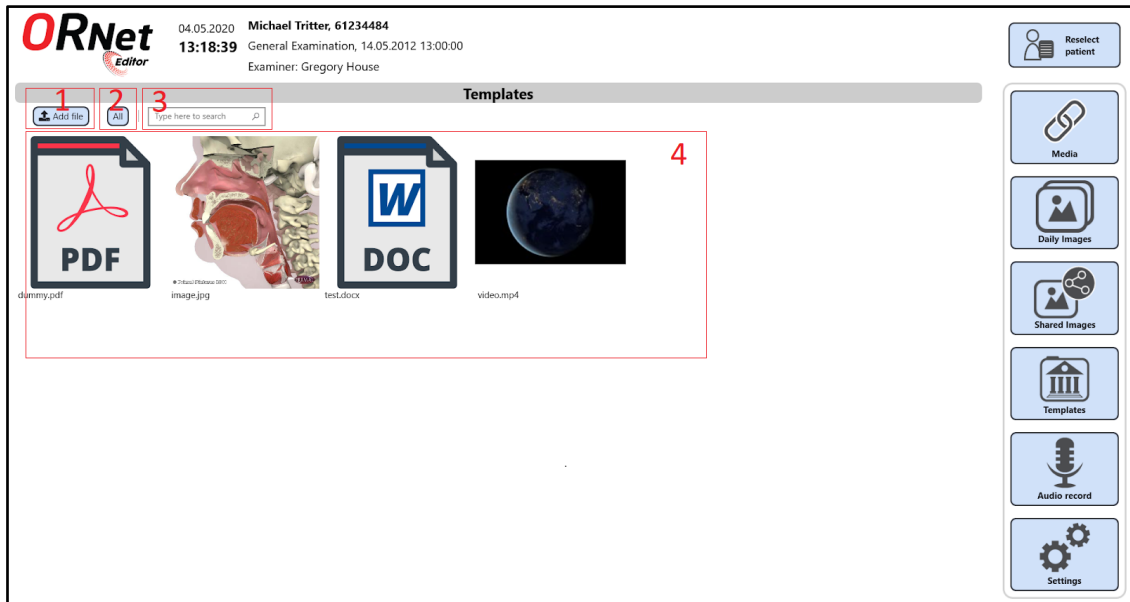


Joonis 18. *Media* vaade näides tume teemaga.

2. *Language* – keelte valimise nuppud.
3. *Containers* – ettevõtte nõuetest tulevad nupud, mis näitavad või peidavad konteinerite ja kassettide informatsioon *Image editor* vaades.
4. *Save changes* – nupp kõike seadiste aktiveerimiseks.

## Templates

Vaade kuvab kõik rakendusele lisatud mustrid. Siin saab mustreid lisada, filtreerida ja otsida.



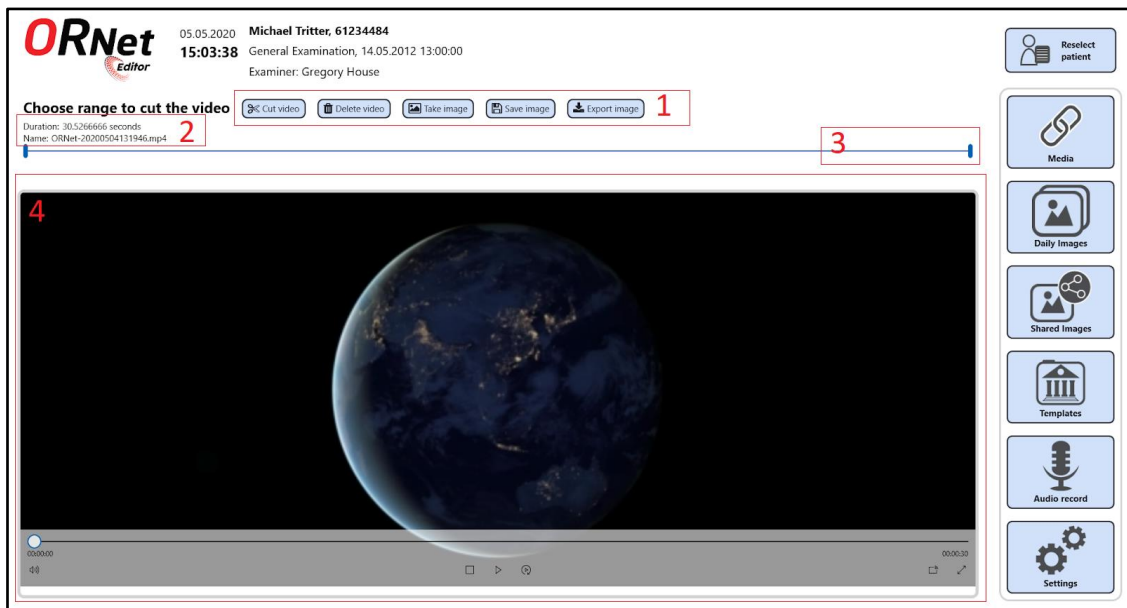
Joonis 19. Templates vaade(rakenduse mustrid)

1. *Add file* – nupp uue mustri lisamiseks(pilt, video, dokument või pdf-fail).
2. *Filter* – nupp failide filtreerimiseks tüüpi järgi.
3. *Search* – otsing nimetuse järgi.
4. *Templates files* – mustrite nimekiri.

## Video editor

Video redigeerimis vaade kus saab videot vaadata, kustutada, videost pildi teha ja valitud ajavahemiku lõigata ja salvestada eraldi failina.



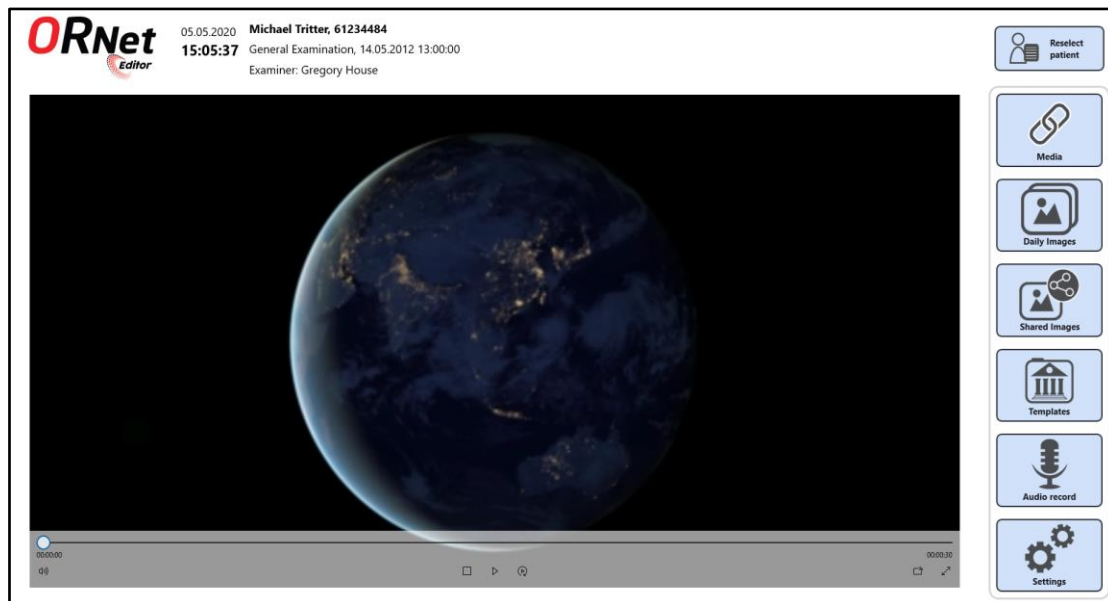


Joonis 20. Video editor vaade(video redigeerimine).

1. *Cut video* – nupp lõigab videost määratud fragmendi ja salvestab eraldi failina.
2. *Delete video* – nupp video faili kustutamiseks.
3. *Take image* – nupp videost pildi tegemiseks.
4. *Save image* – tehtud piltide salvestamine.
5. *Export image* – tehtud piltide salvestamine väljaspool rakenduse keskkonnast.
6. *Info about Video* – video faili nimetus ja kestus.
7. *Cut interval* – menüü lõikamis fragmendi määramiseks.
8. *Video player* – rakenduse sisene video pleier.

### **Video viewer**

Rakenduse sisene video pleier.



Joonis 21. Video viewer vaade(video vaatamine rakenduses).

### 3.5 Kood

Projekti arendamises oli peamiselt kasutatud *C#* programmeerimiskeel. Kasutajaliidese objektide defineerimiseks oli kasutatud *XAML* ja andmebaaside tabelitest andmete kättesaamiseks oli kasutatud *SQL* andmebaasi päringukeel.

Suhtlemine andmebaasiga rakenduses käib *Database* klassi abil, mis asub projektis *Data*.

Meetod *ExecuteQuery* võtab parameetrina päring ja täidab selle päringu andmebaasis. Ja meetod *ExecuteDatabaseConnection* on abi meetod, mis võtab parameetriteks funktsioon ja päring. Sarnaselt eelmise meetodiga luuakse andmebaasiga ühendus. Aga nüüd kasutatakse ka *SqlReader*, mis aitab andmebaasi vastust lugeda.

Päringu genereerimiseks on olemas projektis *Aids* olevad klassid *DatabaseStrings* ja *StringGenerator*.

```

public static void ExecuteQuery(string query)
{
    try
    {
        using (SqlConnection conn = new SqlConnection(DatabaseStrings.GetConnectionString()))
        {
            conn.Open();
            if (conn.State != System.Data.ConnectionState.Open) return;
            using (SqlCommand cmd = conn.CreateCommand())
            {
                cmd.CommandText = query;
                cmd.ExecuteNonQuery();
            }
        }
    }
    catch (Exception e)
    {
        Debug.WriteLine("Database connection exception: " + e.Message);
    }
}

17 references | desuto, 78 days ago | 1 author, 4 changes
private static TResult ExecuteDatabaseConnection<TResult>(Func<SqlDataReader, TResult> action, string query) {
    try {
        using (SqlConnection conn = new SqlConnection(DatabaseStrings.GetConnectionString())) {
            conn.Open();
            if (conn.State == System.Data.ConnectionState.Open) {
                using (SqlCommand cmd = conn.CreateCommand()) {
                    cmd.CommandText = query;
                    return action(cmd.ExecuteReader());
                }
            }
        }
    }
    catch (Exception e) {
        Debug.WriteLine("Database connection exception: " + e.Message);
    }
    return default;
}
}

```

Joonis 22. Klassi *Database* meetodid andmebaasiga suhtlemiseks.

```

public ShapesFactory() {
    _dict = new Dictionary<Enums.Tools?, Func<UIElement>> {
        {Enums.Tools.Freehand, AddPoint },
        {Enums.Tools.Line, AddLine },
        {Enums.Tools.Arrow, AddArrow },
        {Enums.Tools.Rectangle, AddRectangle },
        {Enums.Tools.Circle, AddCircle },
        {Enums.Tools.Text, AddText }
    };
}

6 references | desuto, 164 days ago | 1 author, 1 change
public UIElement Drawing(Point startPoint, Point endPoint, Enums.Tools? tool, UIElement el) {
    _startPoint = startPoint;
    _endPoint = endPoint;
    _activeTool = tool;
    _element = el ?? GenerateBasicElement();
    _dict.TryGetValue(_activeTool ?? throw new InvalidOperationException(), out var createShape (Func<UIElements>));
    var shape UIElement = createShape?.Invoke();
    if (shape != null && !(shape.RenderTransform is CompositeTransform)) {
        shape.RenderTransform = new CompositeTransform();
    }
    return shape;
}
}

```

Joonis 23. Klassi *Shapes Factory* meetodid objektide joonistamiseks.

Meetod *Drawing* kasutatakse pildi redigeerimisel siis, kui kasutaja vasaku nuppu vajutades viib hiire üle tööala (peab olema valitud sobiv tööriist). Selle meetodi abil luuakse algne objekt (kui objekt on juba loodud, siis seda redigeeritakse), mille järel valitakse *Dictionary* kaudu sobiv meetod sõltuvalt objektist, mida soovite joonistada.

Enne projekti alustamist oli kokkulepitud, et me jälgime puhas koodi printsiibi. Kokkulepitud reeglid:

- Kõik parameetrite, meetodite, klasside nimetused ja kommentaarid on inglise keeles.
- Meetodite nimetused on tegusõnad, kui see on võimalik.
- Kõik parameetrid, meetodid ja klasside nimetused on suure algustähega välja arvatud privaatsed parameetrid ja meetodid.
- Nimetuses sõnade eraldamiseks kasutatakse suur täht(nt SuurTäht).

## 3.6 Testid

### Ühiktestid

Testidega on kaetud vähem kui 20% kogu koodi. Kuna eesmärk oli saavutada olemasoleva rakenduse funktsionaalsus lühike aja perioodi jooksul, siis ühiktestide kirjutamine oli osaliselt eiratud.

Ainuke projekt, mis on täielikult testitud on *Aids*. Kuna selles projektis asub rakenduse elementaarne loogika ja abiklassid, siis meie jaoks oli väga oluline, et kõik selle projekti meetodid töötaksid õigesti ja annaksid tulemuse, mida neilt oodatakse. Selle projekti testimiseks kasutatakse testjuhtumid nii standardsete kui ka mitte standardsete sisendandmetega ja nullväärtusega.

### Funktsionaalne testimine

Rakenduse testimiseks oli väga palju kasutatud nii öeldes funktsionaalne testimine, kui erineva stsenaariumitega oli rakendus testitud kasutajaliidese objektide abil. Kasutasime seda meetodit iga päev, kontrollides, kas meie rakendus vastab kõigile ettevõtte nõuetele. Samuti sprindi lõpus ehk kord kuus, siis kui näitasime oma vahetulemused ettevõttele, kellega samamoodi tegime kasutajaliidese ülevaade ja vastavuse kontroll. Kõik tuvastatud probleemid lisasime ülesannete nimekirjale ja parandasime hiljem.

▲	✖ Tests (362)	7 sec
▲	✔ Tests.Aids (144)	7 sec
▷	✔ AudioRecordHe...	61 ms
▷	✔ CacheTests (15)	48 ms
▷	✔ ConverterTests (...)	< 1 ms
▷	✔ DatabaseString...	4 ms
▷	✔ DictionariesTests	13 ms
▷	✔ EnumsTests (19)	3 ms
▷	✔ FileHelperTests (..	6 sec
▷	✔ GetClassTests (5)	1 ms
▷	✔ GetSolutionTests	5 ms
▷	✔ GetZoomFactor...	1 ms
▷	✔ IsAidsTested (1)	3 ms
▷	✔ MathHelperTests	2 ms
▷	✔ MediaEditingHe...	< 1 ms
▷	✔ NotificationsTests	14 ms
▷	✔ PrintHelperTests	< 1 ms
▷	✔ PublicBindingFl...	1 ms
▷	✔ StringGenerator...	< 1 ms
▷	✖ Tests.Aids.Convert...	9 ms
▷	✖ Tests.Data (33)	3 ms
▷	✖ Tests.Domain (1)	2 ms
▷	✖ Tests.Domain.Mo...	2 ms
▷	✖ Tests.Domain.Seri...	3 ms
▷	✖ Tests.Drawing (68)	10 ms
▷	✖ Tests.Facade (1)	2 ms
▷	✖ Tests.Facade.Entiti...	3 ms
▷	⚠ Tests.Facade.Fact...	

Joonis 24. Testide ülevaade.

## 4 Analüüs ja järeldused

### 4.1 Osas 3 esitatud tehnilise teostuse põhjendus

#### Miks just sellised nõuded?

Kõik nõuded olid ettevõtte poolt ette antud, kuigi mõned funktsionaalsed nõuded tekkisid veel protsessi käigus. Samuti oli platvorm ja disain etteantud. Peamine nõue aga oli teisendada olemasolev funktsionaalsust *UWP* platvormile. Need nõuded on olemasoleva rakenduse funktsionaalsus.

Üks väga oluline nõude, mis oli defineeritud aga jäi tegemata on konsultatsiooni võimalus rakenduste vahel. Tagamõtte oli selles, et haigla töötajad saaksid konsulteerida omavahel rakenduse abil. Selline konsultatsioon sisaldab nii audio, kui ka video sidet. Jäi see ülesanne tegemata, sest keskendusime teistedel ülesannetel ja ei jäänud meil piisavalt palju aega ja teadmisi selle ülesanne realiseerimiseks. Küll aga olime uurinud mitmed võimalikud variandid, kui ka olemasoleva rakenduse lähtekoodi, aga realiseerimine jäi tegemata.

#### Miks selline arhitektuur ja millised olid alternatiivid ja miks just selline arhitektuur?

Selline rakenduse arhitektuur oli kasutatud sellepärast, et jagada kogu rakenduse koodi kihtideks. Sellise arhitektuuriga olime mõlemad juba kokku puutunud õppimis protsessi käigus ja meie arvates mitmekihiline arhitektuur aitab koodi lihtsamini lugeda ja taaskasutada. Andmebaasi arhitektuur nagu oli juba mainitud oli etteantud ja uusi tabeleid me ei lisanud.

Arendus protsessi alguses õppisime väga palju uut informatsiooni platvormi kohta ja rakenduse arhitektuurist oli kogemata unustanud. Küll aga siis kui selgus, et on juba palju klassi, mis on ühe tüübiga(näiteks abiklassid, mudelid) hakkasime neid kihtideks jagama. Läks meil pool päeva selleks, et kogu koodi struktureerida ja edaspidi juba kasutasime

struktureeritud arhitektuuri. Sellise arhitektuuriga oli palju lihtsam uut koodi lisada, mis edaspidi kiirendas kogu meie arendamise protsessi.

### **Miks selline disain, milliseid mustreid kasutati ja miks?**

Disain oli etteantud ettevõtte poolt ning vastavalt sellele alternatiiv variante ei ole me otsinud. Samuti ka kasutajaliides oli enim olemas. Kõik meie väikesed muudatused kasutajaliidese disainis olid ettevõttega ette räägitud ja mittevastavuse korral ka muudetud.

### **Millised ja miks olid koodi kirjutamise reeglid?**

Koodi kirjutamise reeglid olid kokkulepitud projekti alguses. Selleks, et igauks juba algusest hakkas neid jälgima selleks, et kood oleks ühtlane. Reeglite valimiseks kasutasime omandatud teadmised ülikoolis puhas koodi kohta ja ka oma harjumused ja eelistused. Reeglite jälgimiseks oli kasutatud *ReSharper*, mis annab hoiatuse reeglite rikkumise korral.

### **Millised ja miks olid testide tegemise meetodid (näidata ka testidega kaetust ja muid meetrikaid)?**

Nagu oli juba mainitud ainuke projekt, mis on lõpuni testitud on *Aids*. See projekt sisaldab abi klassid ja elementaar loogikat. Kõigepealt see oli tehtud sellepärast, et ei oleks meie baas kihis vigu. Peamiselt on need ühiktestid, mis testivad meetodite tulemuse teatud sisendväärtusega. Sealhulgas ka *null* väärtused.

Testidega on kaetud vähem kui 20% kogu koodist. Sest eesmärk oli saavutada olemasoleva rakenduse funktsionaalsus lühike aja perioodi jooksul ja testide kirjutamine oli eiratud. Aga rakenduse testimiseks oli väga palju kasutatud nii öeldes funktsionaalne testimine, kui erineva stsenaariumitega oli rakendus testitud kasutajaliidese objektide abil.

## **4.2 Kirjanduse ülevaade**

### **Uurimisküsimused (RQ)**

- *UWP* eelised ja puudused.

- Kaasaegne meedia failide redigeerimise tarkvara
- *MVVM* arhitektuuri eelised ja puudused

### **Uurimis protsess:**

Artikleid otsitakse kasutades märksõnad nii suurtes otsingusüsteemis kui ka populaarsetel portaalidel teadusartiklitega. Artiklite valimisel oli arvesse võetud artikkelei publitseerimise kuupäev (see peab olema viimase viie aasta jooksul) ja artikkelei vastavus uuringu küsimusele.

Tulemuseks oli leitud rohkem kui 100 artikkelei, mis vastavad meie uuringu küsimustele. Aga pärast lühikokkuvõte lugemist oli valitud kolm kõige sobilikumad artikkelei iga küsimuse peale. Iga valitud artikkel oli põhjalikult uuritud.

### ***RQ1 “UWP eelised ja puudused”***

**Eelis 1.** *UWP*-rakendus saab adapteerida seadme spetsifikatsioonile. [1] [2] [3]

Kõik kaasaegsed *Windows* operatsioonisüsteemid kasutavad ühte tuuma. Tänu sellele arendajad ei pea looma erinevad rakenduste versioonid igale seadmele. Selle asemel rakendus mis on kirjutatud *UWP* platvormil töötab kõikidel seadmetel kuhu on paigaldatud *Windows*-i viimased versioonid: *PC, Tablets, Phone, Xbox, Surface Hub, VR HoloLens, IoT*.

**Eelis 2.** *UWP* toetab mitut programmeerimiskeelt: *Visual C++, C#, VB, JavaScript, XAML, DirectX, HTML, CSS*. [1] [2] [3]

Arendajad saavad kasutada programmeerimiskeele, mis on nendele kõige mugavam või tuttav.

**Eelis 3.** *MVVM*-i arhitektuur on *UWP*-le levinud. [1]

Kõige levinud arhitektuur *MVVM* on kõige efektiivne arhitektuur rakenduse loomiseks (miks just nii on pikemalt kirjeldatud *RQ3*).

**Eelis 4.** *UWP* vajab vähem administratiivset tuge. [1]



Kuna *UWP* rakendus töötab erinevatel seadmetel siis ei ole rohkem vajadust hallata mitu erinevaid versioone. Järelikult märkavalt vähenevad halduskulusid. Samuti kasutajate kogemus jääb stabiilseks sõltumata sellest, millist platvormi kasutaja rakendust kasutab, nii et kasutajaliidese arendamine on palju lihtsam

**Eelis 5.** *Xamarin* koodi on lihtne *UWP*-koodi teisendada. [2]

*Xamarin Code* tavaliselt kasutatakse rakenduse loomiseks *Android/iOS* operatsioonisüsteemidel. Mitmed mobiiltelefoni rakenduse arendajad ei soovi luua rakendused *UWP* platvormil, sest *Windows* operatsioonisüsteemiga mobiiltelefoni on turul märgatavalt vähem kui teisi. Aga *Microsoft*-i poolt tehtud arendusvahendid võimaldavad automaatselt genereerida 90% kogu “*UWP*” rakendusest *Xamarin Code*-i baasil.

**Puudus 1.** *UWP* rakendus ei tööta vanadel operatsioonisüsteemidel (*Windows 8.1*, *Windows 8*, *Windows 7*...). [1] [2]

Kahjuks on võimatu käivitada *UWP* rakendus vanadel operatsioonisüsteemidel. Samuti rakendused suunatud vanadele operatsioonisüsteemidele vajavad modifitseerimist selleks, et olla töövõimelised uuel *UWP* platvormil.

## **RQ2 “Kaasaegne meedia failide redigeerimise tarkvara võimalused (tööriistad)”**

Tänapäeval on olemas erinevad rakendused meedia failide redigeerimiseks, mida saab jagada kolmeks kihtideks [4] [5] [6]:

- *Professional* – rakendused, millistel on lai valik tööriistu, mis võimaldavad teha väga pisikesed muudatused. Näiteks: *Adobe Photoshop*, *DxO PhotoLab*, *Skylum Luminar*
- *Amateur* – rakendused, kus tööriistade kogus on keskmine. Näiteks: *Paint*, *ORNet.Editor*.
- *Base* – rakendused, mis omavad väga piiratud funktsionaalsusega. Kõige sagedamini kasutatakse piltide redigeerimiseks mobiiltelefonil.

Kõige levinumad piltide redigeerimise tööriistad on: *Layers*, *Color filters*, *Gradient*, *Brightness filter*, *Brush/Pen*, *Text*, *Geometric figures*, *Eraser*, *Lasso*, *Ruler*, *Zoom*.

### **RQ3 “MVVM arhitektuuri eelised ja puudused”**

#### **Eelis 1** Täiustatud UX. [7] [8]

Rakendus mis kasutab MVVM arhitektuuri kohaneb kasutaja seadmetega. Järelikult kasutajaliides adapteerub ekraani suurusele või brauseri aknale, mis teeb kasutajaliidese kasutajale sõbralikumaks.

#### **Eelis 2.** Koostöö on lihtsam. [7] [8]

Kuna kasutajaliidese disain ja *Back-end* asuvad üksteisest eraldi, siis tekib võimalus seda ka eraldi areneda. Kusjuures iga arendaja saab tegeleda oma osadega.

#### **Eelis 3.** Uuendamise lihtsus [7] [8] [9]

Tavaliselt on loogika või kasutajaliidese uuendamine vajab muudatused kogu rakenduses. Aga kuna *MVVM* arhitektuuris on nad eraldi osad, siis muudatused ühel osal vajavad vähem muudatusi teisel osal. Järelikult vead ühes osas ei mõjuta teist osa.

#### **Eelis 4.** Testimise lihtsus [7] [9]

Kui *MVVM* arhitektuur on õigesti implementeeritud ja põhiloogika eraldatud teistest koodi osadest. Järelikult testide kirjutamine protsess sellele põhiosale on palju lihtsam.

#### **Eelis 5.** Hooldatavuse lihtsus. [7] [8] [9]

*MVVM* arhitektuuris on koodi osad eraldatud vastavalt ülesandele mida nad teostavad. Järelikult kood on rohkem struktureeritud ja loetav.

**Puudus 1.** *Overkill* lihtsa kasutajaliidese jaoks ja raskesti realiseeritav suurematel juhtudel [8]

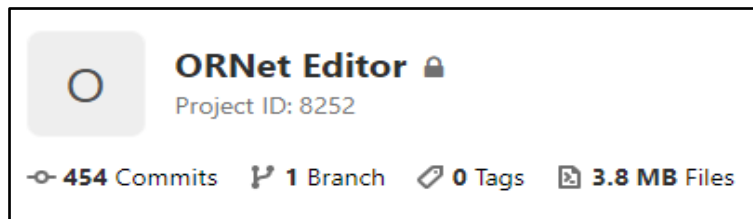
*MVVM* arhitektuuri looja John Grossman ütleb, et *MVVM* arhitektuuri implementeerimine on *overkill* lihtsa kasutajaliidese jaoks. Selle arhitektuuri implementeerimine raskematel rakendustel juba tekib raskused õige disaini loomisel.

## Puudus 2. Sõltub brauserist. [8]

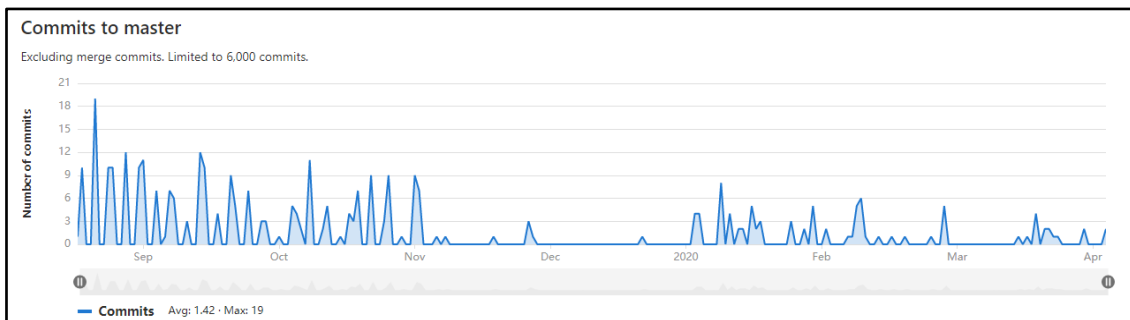
MVVM arhitektuur kasutab brauseri võimalused selleks, et jõudluse paraneda. Mis eeldab, et brauser on *JavaScript-enabled*. Selline funktsioon võimaldab brauseri akna suuruse muuta vastavalt kasutaja seadmele.

### 4.3 Teostatud tööde detailne kirjeldus logi vormis

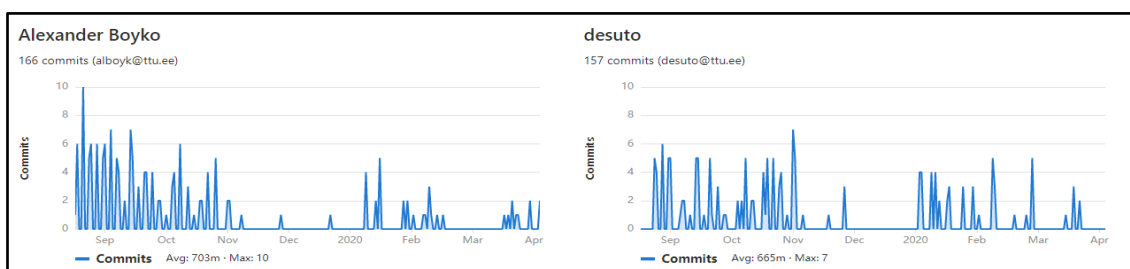
*Gitlab*-i commiti-de väljavõtte:



Joonis 25. Projekti *commit*-ide arv.



Joonis 26. Projekti *commit*-ide arv.



Joonis 27. Projekti *commit*-ide arv.

## Käsitsi koostatud tegevuste logi

Alexander Boyko

09.09.19	Vana projektiga tutvumine	12:30	16:00	3:30:00
12.09.19	Metoodikate uurimine(UWP; MVVM), Andmebaasi ja vana projekti uurimine	9:00	13:00	4:00:00
13.09.19	Projekti loomine, Giti seadistamine, hardkooditud andmetega testimine ja platvormi uurimine	9:00	16:00	7:00:00
16.09.19	Fake andmebaasi tegemine ja andmete filtreerimine	9:00	16:00	7:00:00
19.09.19	Templates vaadega tegelemine, Windows kasutaja nime kuvamine	9:00	15:00	6:00:00
20.09.19	Navigation menu	9:00	15:00	6:00:00
23.09.19	Audio recorder ja audio editeerimine	9:00	15:30	6:30:00
26.09.19	Patsienti raporti vaade tegemine ja võimalus printida raporteid	9:00	16:00	7:00:00
27.09.19	Printimise võimaluse lisamine	9:00	15:00	6:00:00
30.09.19	Patsienti valimine ja filtreerimine	9:00	15:00	6:00:00
03.10.19	Print preview, raporti vaade tegemine	9:00	15:00	6:00:00
04.10.19	Media, shared images, daily images vaaded	9:00	15:30	6:30:00
07.10.19	Rakenduse ühendamine andmebaasiga	9:00	15:00	6:00:00
10.10.19	Fake andmebaasi kustutamine ja andmete andmebaasist võtmine	9:00	15:00	6:00:00
11.10.19	refaktoormine, media filtrid	9:00	14:00	5:00:00
14.10.19	ImageViewer added, filtrid töötavad, UI, add image in media	9:00	15:30	6:30:00
17.10.19	refaktoormine, aids projekti loomine, string generaatori lisamine	9:00	15:00	6:00:00
18.10.19	mediaItemControl	9:00	15:00	6:00:00
21.10.19	refaktoormine reportViewModel uuendamine	9:00	15:00	6:00:00
24.10.19	xaml converters and multiple pages print added	9:00	15:00	6:00:00
25.10.19	pagination and zoom factor added	9:00	15:00	6:00:00
28.10.19	measurement	9:00	14:00	5:00:00
31.10.19	xml writer	9:00	15:00	6:00:00
01.11.19	xml reader	9:00	15:00	6:00:00
04.01.19	Testidega projekti loomine ja kogu rakenduse struktuuri reorganiseerimine	9:00	15:00	6:00:00
8.11.19	refaktoormine	9:00	14:00	5:00:00
11.11.19	XmlHelperi refaktoormine ja AutoTextLabels funktsionaalsus	9:00	15:00	6:00:00
14.11.19	Aids projekti testide kirjutamine	9:00	15:00	6:00:00
15.11.19	Thumbnail refaktoormine	9:00	15:00	6:00:00
18.11.19	bugfixing && PROGRESS RING	9:00	15:00	6:00:00
22.11.19	bugfixing	9:00	15:00	6:00:00
28.11.19	dokumentatsioon	9:00	15:00	6:00:00
29.11.19	dokumentatsioon	9:00	15:00	6:00:00
3.12.19	bugfixing	11:00	15:00	4:00:00
4.12.19	bugfixing	11:00	15:00	4:00:00
12.12.19	audio editor	11:00	15:00	4:00:00
13.12.19	audio editor	11:00	15:00	4:00:00
19.12.19	video editor	11:00	15:00	4:00:00
20.12.19	video editor	11:00	15:00	4:00:00
				219:00:00

Joonis 28. Käsitsi koostatud tegevuste logi.

Date	Description	Start-time	End-Time	Working time
31.01.20	consultation and exercise receiving	10:00:00	12:00:00	2:00:00
03.02.20	setting up project on new machine	12:00:00	15:00:00	3:00:00
03.02.20	localization	15:00:00	18:30:00	3:30:00
03.02.20	localization	17:00:00	23:00:00	6:00:00
04.02.20	localization and new UI design	12:00:00	17:00:00	5:00:00
04.02.20	new UI design	18:00:00	20:00:00	2:00:00
04.02.20	new UI design	20:30:00	23:00:00	2:30:00
06.02.20	bugfixing	13:30:00	14:00:00	0:30:00
07.02.20	meeting	10:00:00	12:00:00	2:00:00
11.02.20	setting logo and splash screen, unauthorized access exception also fixed	14:00:00	17:30:00	3:30:00
11.02.20	fontsize setting	22:00:00	23:30:00	1:30:00
13.02.20	mainpage remake and localization	11:00:00	13:00:00	2:00:00
13.02.20	drag and drop media to compare them later	13:00:00	17:00:00	4:00:00
14.02.20	meeting	10:00:00	12:00:00	2:00:00
21.02.20	patient report displays media name and data, also delete button added	12:00:00	16:00:00	4:00:00
21.02.20	media filters moved to two buttons, also added searchbox	17:00:00	18:00:00	1:00:00
21.02.20	report now show images from different examinations	20:30:00	23:00:00	2:30:00
25.02.20	localization, filters reworked, patient report works with multiple examinations, blue border to video media type added	11:00:00	15:30:00	4:30:00
27.02.20	take snapshot from video	17:30:00	21:00:00	3:30:00
28.02.20	meeting	10:00:00	12:00:00	2:00:00
29.02.20	cut thumbnail from up and bottom, make id, created date etc data be in one column, checkbox added	20:00:00	23:00:00	3:00:00
03.03.20	add doc and pdf file type to templates, search in templates view	17:00:00	20:00:00	3:00:00
05.03.20	pdf viewer, doc opener, snapshot can now be saved to db	19:00:00	23:30:00	4:30:00
09.03.20	icons	16:00:00	21:00:00	5:00:00
10.03.20	bugfixing	15:00:00	17:00:00	2:00:00
13.03.20	meeting	10:00:00	12:00:00	2:00:00
15.03.20	tiny changes	10:30:00	13:00:00	2:30:00
16.03.20	templates video	12:50:00	14:20:00	1:30:00
20.03.20	RTC	18:00:00	23:30:00	5:30:00
21.03.20	RTC	11:00:00	13:00:00	2:00:00
21.03.20	RTC	21:00:00	23:00:00	2:00:00
22.03.20	RTC	19:00:00	23:00:00	4:00:00
23.03.20	RTC	17:00:00	19:00:00	2:00:00
24.03.20	RTC	12:00:00	15:00:00	3:00:00

Joonis 29. Käsitsi koostatud tegevuste logi.

25.03.20	RTC	14:00:00	16:00:00	2:00:00
03.04.20	RTC	9:00:00	18:00:00	9:00:00
16.04.20	RTC	13:30:00	19:30:00	6:00:00
18.04.20	Display or hide completed media	10:30:00	16:30:00	6:00:00
20.04.20	Archive UI	13:00:00	15:00:00	2:00:00
21.04.20	bugfixing	13:00:00	15:00:00	2:00:00
22.04.20	bugfixing	13:00:00	19:00:00	6:00:00
23.04.20	meeting	13:00:00	17:00:00	4:00:00
29.04.20	aftermeeting changes	10:30:00	13:00:00	2:30:00
29.04.20	RTC chat	14:00:00	17:00:00	3:00:00
30.04.20	UDP connection	11:00:00	13:00:00	2:00:00
01.05.20	documentation	10:00:00	16:00:00	6:00:00
04.05.20	tests	12:00:00	14:00:00	2:00:00
04.05.20	documentation	14:00:00	20:00:00	6:00:00
05.05.20	documentation	12:00:00	18:00:00	6:00:00
08.05.20	documentation	12:00:00	16:00:00	4:00:00
11.05.20	documentation	11:00:00	18:00:00	7:00:00

Joonis 30. Käsitsi koostatud tegevuste logi.

Total time	174:30:00
------------	-----------

Joonis 31. Käsitsi koostatud tegevuste logi.

## Deniss Šutov:

1	09.09.19	Learning old app/code/architecture	12:30	16:00	3:30:00
2	12.09.19	Preparation before creating base project.	9:00	13:00	4:00:00
3	13.09.19	Base-Project creation.	9:00	16:00	7:00:00
4	16.09.19	FakeDB, Enums,	9:00	16:00	7:00:00
5	19.09.19	ImageEditor View created, base-xaml view components, base functions	9:00	15:00	6:00:00
6	20.09.19	Image now showing in Image editor, Base Undo Redo Saving functionality added	9:00	15:00	6:00:00
7	23.09.19	Image-saver is working, testing some functionality	9:00	15:30	6:30:00
8	26.09.19	ColorPicker, Custom tools, Zooming	9:00	16:00	7:00:00
9	27.09.19	Drawing line with InkCanvas.	9:00	15:00	6:00:00
10	30.09.19	Massive rework InkCanvas -> Canvas	9:00	15:00	6:00:00
11	03.10.19	Started working on selection, bugfixing	9:00	15:00	6:00:00
12	04.10.19	Rectangle, ellipse drawing working	9:00	15:30	6:30:00
13	07.10.19	Fixing bugs in image editor	9:00	15:00	6:00:00
14	10.10.19	Fixing bugs, working on drag and drop functionality (Manipulations)	9:00	15:00	6:00:00
15	11.10.19	Refactoring, working on Drawing object's resizing.	9:00	14:00	5:00:00
16	14.10.19	Working on manipulations continue	9:00	15:30	6:30:00
17	17.10.19	Manipulations... + project structure enhancing	9:00	15:00	6:00:00
18	18.10.19	Prepating for XML reading/writing, refactoring	9:00	15:00	6:00:00
19	21.10.19	Selection functionality ready	9:00	15:00	6:00:00
20	24.10.19	Working on resize functionality	9:00	15:00	6:00:00
21	25.10.19	refactoring	9:00	15:00	6:00:00
22	28.10.19	Text-adding functionality	9:00	14:00	5:00:00
23	31.10.19	refactoring	9:00	15:00	6:00:00
24	01.11.19	Container created, Mesurement functions	9:00	15:00	6:00:00
25	04.01.19	Working with Container, Tests project added	9:00	15:00	6:00:00
26	07.01.19	Working on Arrow	9:00	15:00	6:00:00
27	8.11.19	DrawingProjects refactoring, Auto-text, Text-Frame added	9:00	14:00	5:00:00
28	14.11.19	refactoring Moving/resizing, trying to do rotation	9:00	15:00	6:00:00
29	15.11.19	Major bug-fixing	9:00	15:00	6:00:00
30	18.11.19	Cursor-changing, bug-fixing	9:00	15:00	6:00:00
31	21.11.19	Arrow-bugs fixing, Cursor-bug fixing	9:00	15:00	6:00:00
32	22.11.19	Calibration, autoText, TextSelection BoxBlock converter	9:00	15:00	6:00:00
33	28.11.2019	Exceptions fixing, bug fixing, adding tests	9:00	15:00	6:00:00
34	29.11.19	Bug-fixing	9:00	15:00	6:00:00
35					
36					201:00:00

Joonis 32. Käsitsi koostatud tegevuste logi.

	A	B	C	D	E
1	Date	Description	Start-time	End-Time	Working time
2	31.01.20	Получение задания, постановка задач, составление TODO листа	10:00	12:00	2:00:00
3	31.01.20	Added "Draw a Line!" message after pressing calibration button	21:10	21:30	0:20:00
4	31.01.20	Drawing Controller Merged with ImageEditor. Text-Adding bug fixed	21:30	22:20	0:50:00
5	31.01.20	ImageEditor refactoring	22:20	23:15	0:55:00
6	31.01.20	DrawingElement Merged with DrawingStroke + refactoring + Text BugFixed	23:15	23:55	0:40:00
7	01.02.20	After selecting drawing object, current Color/Thickness/Textsize is changed to object's properties.	17:00	18:00	1:00:00
8	01.02.20	Models added: Workstation, Container, ContainerType, ExaminationContainers, Added fields to Media model	18:00	18:25	0:25:00
9	01.02.20	Measurements Text color is now the same as figure color	18:25	18:30	0:05:00
10	01.02.20	Added information Flyout	18:30	18:45	0:15:00
11	06.02.20	Added ContainerNumber and CassetteNumber to ImageEditor Page	12:00	13:30	1:30:00
12	06.02.20	Understanding Database Architecture	13:30	14:30	1:00:00
13	06.02.20	Meeting preparation	14:30	15:00	0:30:00
14	06.02.20	Drawing classes refactoring + UNDO/REDO for manipulations started	15:30	16:50	1:20:00
15	06.02.20	UndoRedoSerie Rework, ImageEditor Refactoring	17:50	21:40	3:50:00
16	07.02.20	Meeting	10:00	12:00	2:00:00
17	08.02.20	Icons Added. ImageEditor icons changed.	14:50	15:30	0:40:00
18	08.02.20	Working on ImageEditor UX/UI improvement	18:40	20:20	1:40:00
19	08.02.20	ImageEditor refactoring, adjustments, Drawing Bug Fixed.	20:20	22:30	2:10:00
20	10.02.20	Icon Bugfix	17:50	18:00	0:10:00
21	10.02.20	Installer	18:50	19:20	0:30:00
22	10.02.20	IE bugfixes	19:20	21:00	1:40:00
23	14.02.20	Meeting	10:00	12:30	2:30:00
24	15.02.20	ContainerCassettes development	17:00	21:10	4:10:00
25	15.02.20	fixing Line Arrow resize bug	21:10	22:20	1:10:00
26	15.02.20	Bugfixing	22:20	22:50	0:30:00
27	17.02.20	Bugfixing	21:10	21:50	0:40:00
28	22.02.20	Container Cassettes functionality	19:00	21:00	2:00:00
29	27.02.20	Arrow resizing bugfix	13:00	15:00	2:00:00
30	27.02.20	Cassette label cannot longer be edited	15:00	17:00	2:00:00
31	28.02.20	meeting	10:00	12:00	2:00:00
32	06.03.20	meeting	10:00	11:00	1:00:00
33	08.03.20	Arrow resize rework	17:30	20:20	2:50:00
34	08.03.20	Resize Total rework	20:20	22:40	2:20:00
35	09.03.20	Resize rework #2 + bugfixing + refactoring	16:00	23:00	7:00:00
36	13.03.20	Meeting	10:00	12:00	2:00:00
37	19.03.20	Working on rotation	18:00	21:00	3:00:00

Joonis 33. Kätsi koostatud tegevuste logi.

37	19.03.20	Working on rotation	18:00	21:00	3:00:00
38	25.03.20	Consultations, XML for Settings	12:00	19:00	7:00:00
39	26.03.20	working on XML	12:30	14:00	1:30:00
40	27.03.20	working on SettingsPropertiesXML	12:00	16:00	4:00:00
41	28.03.20	SettingsProperties reader is ready	12:30	14:00	1:30:00
42	28.03.20	SettingsProperties reader bugfix + enhancements	14:00	15:30	1:30:00
43	28.03.20	IE refactoring	15:30	16:00	0:30:00
44	31.03.20	IE refactoring	12:30	13:30	1:00:00
45	04.04.20	refactoring	12:00	15:00	3:00:00
46	07.04.20	Restore project	18:00	1:00	7:00:00
47	10.04.20	Containers	12:00	14:00	2:00:00
48	14.04.20	Containers, refactoring	16:00	21:00	5:00:00
49	18.04.20	Containers	14:00	20:00	6:00:00
50	19.04.20	Containers/bugfix	17:00	19:00	2:00:00
51	20.04.20	Installer	18:00	20:00	2:00:00
52	21.04.20	Installer	16:00	23:00	7:00:00
53	04.05.20	documentation	15:00	20:00	5:00:00
54	05.05.20	documentation	17:00	21:00	4:00:00
55	07.05.20	documentation	16:00	20:00	4:00:00
56	09.05.20	documentation	14:00	23:00	9:00:00
57	10.05.20	documentation	13:00	25:00:00	12:00:00

Joonis 34. Kätsi koostatud tegevuste logi.

Total time	143:40:00
------------	-----------

Joonis 35. Kätsi koostatud tegevuste logi.

## 4.4 Hinnang projekti teostamise protsessi kohta

Projekt oli alustatud 2019 aastal septembri keskel ja kursuse *ITB1706 - Infosüsteemide arendamise meeskonnaprojekt* raames oli see edukalt kaitstud detsembris. Projekti arendamisega jätkasime juba jaanuaris uuel semestril lõputöö raames. Ehk projekti esimese osa teostati septembrist detsembrini ja teine osa oli teostatud jaanuarist maini.

Ettevõtte poolt projekti juhendasid: Inga Moreva, Brigita Urmet ja Jarmo Poolak. Ja ülikooli poolt: Jekaterina Tšukrejeva ja Kristina Murtazin.

Suhtlemine nii ettevõttega, ülikooli juhatajatega kui ka omavahel toimus meili kaudu. Ettevõttele esitasime oma vahetulemused kontoris, nendel koosolekutel saime kõige rohkem ülesandeid ja tagasisidet. Esimesel poolel tegime tööd ettevõtte kontoris, mis väga lihtsustas suhtlemise. Aga teisel poolel juba tegime tööd kodus, alguses selle tõttu, et kõik ülesanded oli meie jaoks selged ja tegutsesime arendamisega. Pärast aga juba eriolukorra kehtestamise tõttu.

Ülesannete jagamine toimus vastavalt ülesannete nimekirjale. Detailne kirjeldus on olemas peatükis tööprotsessi kirjeldus.

Meie jaoks oli suur eelis, et projekt kirjutatakse *C#* programmeerimis keeles, kuna oleme sellega juba varem kokku puutunud. Aga sellest platvormidest nagu *UWP* ja *WPF* oli vaja alguses väga palju juurde õppida. Samuti kasutajaliidese objektide defineerimine *XAML* keeles oli meie jaoks uus asi. Olime väga palju lugenud *Microsoft*-i poolt tehtud dokumentatsiooni nende platvormide kohta, mis projekti alguses võttis kõige suurema ajakulu.

Kõikide koodi kirjutamisega seotud probleemidega tegelesime ainult iseseisvalt. Me küll saime kasutada olemasoleva koodi, aga paljudel juhtudel platvormi erinevus ei lubanud seda teha. Samamoodi saime ka ettevõtte poolt tagasiside koosolekul rakenduse kohta, aga need märkused olid pigem seotud kasutajaliidese disainiga. Ühelt poolt see oli kasulik, sest saaksime probleemidega ise hakkama saada. Teiselt poolt see võttis rohkem aega, kui me alguses plaanisime. Mis tõi tulemuseks mõned funktsionaalsuse nõuded tegemata.

Ettepanekuks tahame öelda, et oleks väga kasulik kui alguses keegi ülikooli poolt koos meiega süvenes projekti probleemile ja aitas teha esimesed sammud. Sest need olid kõige raskemad ja pingelikud.

#### **4.5 Meeskondlik konsensuslik hinnang kõikide meeskonnaliikmete panuse kohta**

Alexander Boyko - 0 - panustas samaväärselt

Deniss Šutov - 0 - panustas samaväärselt



## 5 Kokkuvõte

Kokkuvõtteks võime öelda, et saime töötava *UWP* rakenduse sama funktsionaalsusega, mis on olemasolevas rakenduses ja mis suhtleb olemasoleva andmebaasiga. Kuna *UWP* on käivitata *Windows 10* operatsioonisüsteemis võime öelda, et meie rakendus on kaasaegne.

Eesmärk oli pakkuda kasutajatele kaasaegset, intuitiivset kasutust omavat tarkvara, mis oleks hallatav ja kuhu oleks lihtne uusi omadusi lisada. Võime öelda, et eesmärgid on täidetud pooleli, sest rakendus on tehtud uuel platvormil ja kindla arhitektuuriga, mis võimaldab lihtsasti uusi omadusi lisada ja rakenduse disain on teisendatud, mis annab kasutajaliidese intuitiivsuse. Aga rakendus ei ole veel lõpuni testitud ning puudub konsultatsiooni võimalus.

Projekti alguses tehtud analüüs ja valitud tööriistad ja metoodika aitasid meile arendamise käigus. Kus tööriistad aitasid koodi paremini hallata, agiilne arendusmetoodika aitas meile ülesandeid jagada ja nii enda vahel kui ka ettevõttega suhelda.

Töö tulemusena on saadud töötav rakendus, mis suhtleb olemasoleva andmebaasiga ja osaliselt omab olemasoleva rakenduse funktsionaalsus. Samuti kasutajaliidese disain oli korralikult välja töötatud ja töötav erineva resolutsiooniga arvutitel. Rakenduse mitmekihiline arhitektuur aitab koodi lihtsamini lugeda ja juurde kirjutada. Koodi kirjutamise reeglid säilivat koodi ühtsuse.

Kuna suurem osa nõuetest ja implementeerimis arhitektuurist oli juba ette antud, siis analüüsime ainult enda poolt kirjutatud koodi ja rakenduse arhitektuuri. Mis meie arvates on struktureeritud ja loetav. *UWP* platvormi tehtud analüüs andis meile ülevaade selle platvormi kohta, platvormi eelistest ja puudustest. Kuulutatud aega jooksul saime suurem osa nõudeid valmis.

## Kasutatud kirjandus

- [1] M. Martel, „linkedin.com,“ 9 05 2016. [Võrgumaterjal]. Available: <https://www.linkedin.com/pulse/benefits-uwp-development-universal-windows-platform-michel-martel/>. [Kasutatud 13 05 2020].
- [2] E. Szkiładź, „softwarehut.com,“ 12 09 2017. [Võrgumaterjal]. Available: <https://softwarehut.com/blog/tech/universal-windows-platform-current-state-future>. [Kasutatud 13 05 2020].
- [3] jbe2277, „github.com,“ 1 12 2018. [Võrgumaterjal]. Available: <https://github.com/jbe2277/waf/wiki/UWP-vs.-WPF>. [Kasutatud 13 05 2020].
- [4] R. Lawton, „techradar.com,“ 01 05 2020. [Võrgumaterjal]. Available: <https://www.techradar.com/news/best-photo-editor>. [Kasutatud 13 05 2020].
- [5] A. Bateman, „enviragallery.com,“ 14 04 2020. [Võrgumaterjal]. Available: <https://enviragallery.com/best-photo-editing-software-for-photographers/>. [Kasutatud 13 05 2020].
- [6] M. Muchmore, „pcmag.com,“ 08 04 2020. [Võrgumaterjal]. Available: <https://www.pcmag.com/picks/the-best-photo-editing-software>. [Kasutatud 13 05 2020].
- [7] XiaoLong Li, DaLiang Chang, Hui Pen, XiaoYu Zhang, YuanXin Liu, YaXian Yao, „ieeexplore.ieee.org,“ 10 06 2015. [Võrgumaterjal]. Available: <https://ieeexplore.ieee.org/document/7288144>. [Kasutatud 13 05 2020].
- [8] J. Bonniwell, „sagitec.com,“ 25 07 2018. [Võrgumaterjal]. Available: <https://www.sagitec.com/blog/the-5ws-of-mvvm>. [Kasutatud 13 05 2020].
- [9] BlogsNook, „blogsnook.com/,“ 8 05 2019. [Võrgumaterjal]. Available: <https://blogsnook.com/mvvm-pattern-advantages/>. [Kasutatud 13 05 2020].
- [10] M. Inc., „docs.microsoft.com,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/windows/uwp/>. [Kasutatud 13 05 2020].
- [11] „en.wikipedia.org,“ [Võrgumaterjal]. Available: <https://en.wikipedia.org/wiki/SQL>. [Kasutatud 13 05 2020].
- [12] „en.wikipedia.org,“ [Võrgumaterjal]. Available: [https://en.wikipedia.org/wiki/Extensible\\_Application\\_Markup\\_Language](https://en.wikipedia.org/wiki/Extensible_Application_Markup_Language). [Kasutatud 13 05 2020].
- [13] „https://dl.acm.org/,“ [Võrgumaterjal]. Available: <https://dl.acm.org/>. [Kasutatud 13 05 2020].
- [14] „presmarmethuen.org,“ 18 12 2019. [Võrgumaterjal]. Available: <https://presmarmethuen.org/et/dictionary/what-s-the-difference-between-a-net-standard-class-library-and-a-net-framework-class-library/>. [Kasutatud 17 05 2020].

## **Lisa 1 – Enda panuse kirjeldus ja eneseanalüüs – Alexander Boyko**

### **Enda panuse kirjeldus**

Minu peamised ülesanded olid järgmised: kasutajaliidese disain, audio ja video redigeerimine, raportite koostamine, lokaliseerimine, ühendus andmebaasiga. Tegelesin ka vaatega: *Media*, *Templates* ja *Home Page*. Mis sisaldavad loogikat patsiendi, uuringu ja meedia andmete kuvamiseks, filtreerimiseks, otsimiseks, valimiseks ja lisamiseks. Samuti tegelesin *Main Page* ehk navigeerimis vaatega ja kogu navigeerimisega üldiselt. Sealhulgas on põhivaade konstruktorid, mis vastavalt antud parameetritest avanevad keskel valitud vaade. Teisendasin kõik olemasoleva rakenduse ikoonid ja logod ning kasutasin ka nuppude tähistamiseks *UWP* platvormil olevad ikoonid, et disain oleks kasutajale sõbralikum. Lisasin rakendusele ka lokaliseerimise ehk keelte valimise võimaluse ja ka võimaluse uusi keeli lisada. Tegin ka muudatused rakenduse kasutajaliidese disainis, et saaks rakenduse teemat tumedaks muuta. Lindistamisega seotud kood on samuti minu poolt kirjutatud. Ja lisaks ka audio ja video lõikamine ning videost pildi tegemine.

Alguses oli vaja väga palju juurde õppida ja õppimis protsess käis koos koodi kirjutamisega. Alustasin nende ülesandega, mis minu jaoks olid tuttavad. Pärast uurimist ja juurde õppimist juba tegelesin keeruliste ülesandega. Tööprotsess oli igal meie sprinti jooksul väga sarnane. Kõigepealt tegelesin saadud kommentaaridega ja märkustega ettevõtte poolt. Mis mõnikord vajasisid väga suurt muudatust. Pärast juba uurisin informatsiooni oma järgmise ülesande kohta. Siis kui informatsiooni oli kogutud piisavalt palju hakkasin juba kirjutada koodi ja seda testida. Lõpus refaktorisin kirjutatud koodi ja vajadusel ka tegin abiklassid. Kui suurem osa ülesannetes oli tehtud ja sprint lõppeks näitasime oma tulemused ettevõttele ja algas uus sprint.

Protsessi käigus tegelesin ka rakenduse analüüsiga ning pakkusin mõned lahendused. Näiteks oli minu initsiatiiv ümber teha rakenduse disaini ning kasutada mitmekihilise arhitektuuri.

### **Eneseanalüüs**

Kõige keeruline koht minu jaoks oli alguses kui me hakkasime projektiga tutvuma. Olemasolev rakendus, kui ka andmebaas oli meie varasemalt ülikoolis tehtud projektides nii palju suurem, et ei saanud alguses aru millest alustada. Väga aitas aga see, et nii programmeerimiskeel, kui ka töökeskkond olid tuttavad. Samuti oli mugav kahekesi õppida ja töötada, sest jagasime koguaeg õpitud materjal omavahel ja raskuste tekkimisel omavahel arutada.

Minu jaoks oli kogu protsess väga arendatav, sest esimest korda sain süveneda reaalsele arendamis protsessile. Kõik arutlused ja koosolekud olid väga arendatavad. Sain väga palju uut informatsiooni *UWP* kohta, mille peamised infoallikad olid Microsoft-i poolt tehtud dokumentatsioon. Tunnen, et pean tulevikus rohkem keskenduma tellija vajadustele ning küsida rohkem küsimusi.

Väga aitas mulle, et ülikool olime õppinud peamiselt *C#* programmeerimiskeelt ning kasutasime selleks töökeskkonna *Visual Studio*. See aitas meile alustada meie töö mitte keele ja keskkonna vaid juba platvormiga uurimisega. Samuti ei tekkinud mul probleemi andmebaasi struktuuri ja disaini arusaamisega, sest relatsiooniline andmebaas oli minu jaoks juba tuttav.

## Lisa 2 – Enda panuse kirjeldus ja eneseanalüüs – Deniss Šutov

### Enda panuse kirjeldus

Projekti juures tegelesin peamiselt *ImageEditor*'i funktsionaalsusega. Ehkki see on vaid üks paljudest liikidest, on sellel ka suurim funktsionaalsus:

- Pildid avatakse muutamiseks eraldi aknas.
- Võimalus muuta pildi skaalat (sisse suumida, välja suumida, tagastada alussuurus).
- Võimalus pilti muuta, lisades sellele erinevaid kujundeid (joon, nool, ristkülik, ellips, freehand, tekst).
- Igal objektil on oma raam, mis ilmub kuju valimisel. Seda kasutatakse manipulatsioonide mugavamaks rakendamiseks.
- Iga geomeetrilise kujundi jaoks arvutatakse übermõõt (pikkus) ja pindala (kui on) pikslites.
- Kalibreerimine - võimalus seada eriparameeter *PixelToMMRatio*, mis teisendab mõõtmised pikslitest millimeetriteks, kajastades tegelikku suurust.
- Võimalus muuta kuju värvi / jooni paksust / teksti suurust.
- *Undo/Redo* funktsionaalsus.
- Valitud kuju kustutamine.
- Ekspordi pilt - salvestage uus pilt koos kõigi muudatustega.

- Kui avate rakendusest pilte, laaditakse kõik joonised XML-failist. Pildi salvestamisel kirjutatakse XML fail üle. Seega saab kõiki objekte igal ajal kustutada ja muuta.
- *Container-Cassettes* funktsionaalsus: koos kujudega ilmub number, mis näitab mis osa pildist asub kassetis selle numbriga. Pildile ka tehakse seos konteineriga andmebaasis, kui on kasutatud esimeses korras.
- *XML for Settings*
- *Image Editor UI UX*: nuppude asukohad, jne.
- *Bugfixing*
- *Refactoring*
- *Testing*
- *Installer*

## **Eneseanalüüs**

Töö käigus oli minu jaoks suurim väljakutse enesekorraldus. Kui Sügisel Me võiksime meeskonnaprojekti kallal töötades ikkagi tulla ettevõtte kontoris, sealt nõu pidada ja koos töötada, siis kevadel pidime seoses pandeemiaga töötama kodust, mis minu jaoks isiklikult mõjus minu tootlikkusele väga negatiivselt.

Peale selle polnud töös erilisi probleeme. Muidugi oli keerulisemaid hetki, kus pidime mitu päeva töötama. Oli ka lihtsamaid ülesandeid, mille mõne minutiga lõpetasime. Internetis oli aga piisavalt teavet, et saaksime peaaegu iga ülesandega hakkama.

Kõige tähtsam, mida projekti kallal õppisin, on see, et tarkvaraarendus ei seisa paigal. Hiljuti kasutatud arendusriistad võivad nüüd täiesti kasutatud olla. Seega peame pidevalt edasi liikuma ja õppima uusi tehnoloogiaid, vastasel juhul on võimatu luua konkurentsivõimelist tarkvara.

Projekti käigus süvendasin ka oma arusaamist professionaalsuse kontseptsioonist. Igaüks saab kirjutada koodi, mis täidab ülesannet. Kui junior teeb seda aga kuu jooksul ja mitte

parima kvaliteediga, siis teeb professionaal seda nädala jooksul parimal viisil. Küsimus ei ole võimes ülesannet täita, vaid selle ülesande tõhususes.

Usun, et meil olid kõik projekti lõpuleviimiseks vajalikud oskused. Väärrib märkimist, et ka meil vedas projektiga, millega tegelesime. Esiteks kasutati projektis tuttavat programmeerimiskeelt C#. Teiseks oli meil näide rakendusest, mis peaks osutama. Nagu ka head juhid, kellelt võiksime igal ajal abi küsida. Seega polnud projekt (minu jaoks isiklikult) keeruline.