

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Johanna Maria Koplik 185883IABB

Linda Hmelevskaja 185612IABB

**MEESKONNAPROJEKT:
MEDITSIINIRAKENDUSE AKNA
ÜLEVIIMINE UNIVERSAL WINDOWS
PLATVORMILE**

Bakalaureusetöö

Juhendajad: Kristina Murtazin
MSc
Karl-Erik Karu
MSc

Tallinn 2021

Autorideklaratsioon

Kinnitan, et olen 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.

Autor: Johanna Maria Koplik, Linda Hmelevskaja

17.05.2021

Annotatsioon

Meeskonnaprojekti eesmärgiks on Bait Partner OÜ meditsiininirakendusele *Study Info* akna loomine. Meditsiinitarkvara peamiseks kasutajateks on haiglad. Süsteem aitab lihtsustada kirurgia ja patoloogia valdkonna töötajate igapäevaseid protsesse.

Antud lõputöö eesmärgiks on varasemalt autorite poolt arendatud *Study Info* akna üleviimine *Universal Windows* platvormile. Lahendus peab olema sobilik ettevõtte uue tootega. Teiseks eesmärgiks on võrrelda WPF ja UWP lahendust kirjanduse ja enda tehtud koodi põhjal ning analüüsida nende eelised ja puudused. Kolmandaks eesmärgiks koostada WPF-st UWP-le teisaldamise juhend, mis oleks abiks teistele, kes plaanivad uuele platvormile üle minna.

Lõputöö tulemusena valmis aken uuele platvormile olemasoleva funktsionaalsusega ja uuendatud kasutajaliidese disainiga. Valmistati juhend, mis aitaks teisi UWP-le üleminekuks ja analüüsiti erinevusi ning sarnasusi WPF kui ka UWP rakenduste puhul.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 33 leheküljel, 5 peatükki, 34 joonist, 3 tabelit.

Abstract

Team project: Transferring Window for Medical Application to Universal Windows Platform

The aim of the team project is to create a Study Info window for the medical application of Bait Partner OÜ. The main users of medical software are hospitals. The system helps to simplify the daily processes of surgery and pathology workers. For example, software can be used to remotely control devices in a room during surgery or simply take pictures of a sample during a pathological examination.

Based on the feedback from current users, an additional feature is desired and there is need to add a window with general information about the patient's procedure displayed in one place. Information is currently available hidden in each function's own window or application, and the aim of the project is to display the information in one window, which helps to speed up the work process when using medical software, creating added value that would satisfy the need of hospital staff to get summary study information from one place.

The aim of thesis is to transfer the Study Info window previously developed by the authors to the newer platform Universal Windows. The solution must be compatible with the company's new product. Second goal is to compare the WPF and UWP solutions based on the literature and self-developed code to analyze their differences and similarities. Third goal is to put together instructions to transfer projects from WPF to UWP what can be support to others who plan to move to new platform.

As a result of the thesis, a window for the new platform with existing functionality and an updated user interface design was completed. A guide was developed to help others transition to UWP and differences between WPF and UWP were analyzed.

The thesis is in Estonian and contains 33 pages of text, 5 chapters, 34 figures, 3 tables.

Lühendite ja mõistete sõnastik

AAA	<i>Arrange, Act, Assert</i>
API	<i>Application Program Interface</i>
CPU	<i>Central Processing Unit</i>
HTML	<i>HyperText Markup Language</i>
MB	<i>Megabyte</i>
MVVM	<i>Model-View-ViewModel</i>
PACS	<i>Picture Archiving and Communication System</i>
SDK	<i>Software Development Kit</i>
SQL	<i>Structured Query Language</i>
SSN	<i>Social Security Number</i>
UI	<i>User Interface</i>
UWP	<i>Universal Windows Platform</i>
XAML	<i>Extensible Application Markup Language</i>
XML	<i>Extensible Markup Language</i>

Sisukord

1 Sissejuhatus	10
1.1 Probleem ja projekti eesmärk	10
1.2 Lühülevaade teostatud funktsionaalsustest.....	11
1.3 Töö edasine struktuur	11
2 Metoodika.....	13
2.1 Objekti detailne kirjeldus.....	13
2.2 Tööriistade kirjeldus	15
2.3 Tööprotsessi kirjeldus.....	15
3 Tulemused	17
3.1 Süsteemi funktsionaalsed ja mittefunktsionaalsed nõuded	17
3.2 Arhitektuur.....	17
3.3 Disain.....	20
3.4 Kood	20
3.5 Testid	21
4 Analüüs.....	22
4.1 Tehnilise teostuse analüüs	22
4.2 Kirjanduse ülevaade	23
4.3 Teostatud tööde detailne kirjeldus logi vormis.....	31
4.4 Hinnang projekti teostamise protsessi kohta	40
4.5 Meeskondlik hinnang panuse kohta	41
5 Kokkuvõte	42
Kasutatud kirjandus	43
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	45
Lisa 2 – Enda panuse kirjeldus ja eneseanalüüs, Johanna Maria Koplik	46
Lisa 3 – Enda panuse kirjeldus ja eneseanalüüs, Linda Hmelevskaja.....	48
Lisa 4 – <i>Study Info</i> akna vaade	50

Jooniste loetelu

Joonis 1. Projekti arhitektuur.....	18
Joonis 2. Andmebaasi arhitektuur.	19
Joonis 3. Läbitud testid.....	21
Joonis 4. UWP rakenduse XAML <i>Run</i> element.....	25
Joonis 5. WPF rakenduse XAML <i>multibinding</i>	25
Joonis 6. WPF rakenduse XAML nimeruumide importimine.....	26
Joonis 7. UWP rakenduse XAML nimeruumide importimine.....	26
Joonis 8. UWP rakenduse DockPanel näide.....	26
Joonis 9. WPF rakenduse DockPanel näide.	26
Joonis 10. WPF rakenduse mälu kasutamine.	28
Joonis 11. UWP rakenduse mälu kasutamine.....	28
Joonis 12. WPF rakenduse CPU kasutamine.....	28
Joonis 13. UWP rakenduse CPU kasutamine.....	28
Joonis 14. Visual Studio Installeri installatsioon.	29
Joonis 15. Arendaja sätete lubamine.	29
Joonis 16. TCP/IP protokollide lubamine.	30
Joonis 17. <i>Commit</i> -id projekti peale.....	31
Joonis 18. <i>Commit</i> -id Johanna Maria Koplik.....	32
Joonis 19. <i>Commit</i> -id Linda Hmelevskaja.....	32
Joonis 20. Väljavõtte ajakulust 1.....	32
Joonis 21. Väljavõtte ajakulust 2.....	33
Joonis 22. Väljavõtte ajakulust 3.....	34
Joonis 23. Väljavõtte ajakulust 4.....	35
Joonis 24. Väljavõtte ajakulust 5.....	36
Joonis 25. Väljavõtte ajakulust 6.....	37
Joonis 26. Väljavõtte ajakulust 7.....	38
Joonis 27. Väljavõtte ajakulust 8.....	39
Joonis 28. Väljavõtte ajakulust 9.....	40
Joonis 29. <i>Study Info</i> akna vaade 1.....	50

Joonis 30. *Study Info* akna vaade 2..... 51

Tabelite loetelu

Tabel 1. Uuringu üldinformatsiooni väljad.	13
Tabel 2. Patoloogia uuringu väljad.....	14
Tabel 3. Patoloogia uuringuga seotud proovi väljad.	14

1 Sissejuhatus

Bait Partner OÜ on tehnoloogilisi lahendusi ja teenuseid pakkuv ettevõtte, kes on suunatud meditsiiniliste ja audiovisuaalsete erilahenduste arendamisele, integreerimisele ja müügile. Bait Partner on meditsiinitarkvara kaubamärgi omanik ja kõigi selle kaubamärgi toodete tootja. Meditsiinitarkvara on spetsiaalne meditsiinilahendus operatsioonisaali seadmete haldamiseks, integreerimiseks, meditsiiniliseks pildistamiseks, patoloogiliste tööde kuvamiseks ning telepatoloogia juhtimissüsteemiks. Integreerides infotehnoloogia, pildindus ja meditsiin ühte võrku lihtsustatakse meditsiinitöötajate töövoogu ja suurendatakse patsientide ohutust [1].

Meeskonnaprojekti aine raames valmis meditsiinitarkvarale aken *Study info*, mille arendasid välja enda tehtud prototüüpi põhjal Tallinna Tehnikaülikooli 3. kursuse äriinfotehnoloogia tudengid Johanna Maria Koplik, Linda Hmelevskaja, Aleksandr Levitsev ja Raul Vilippus. Käesoleva töö raames seati aken UWP (*Universal Windows Platform*) platvormile ning tehti täiendused valminud projektile.

1.1 Probleem ja projekti eesmärk

Meditsiinitarkvara peamiseks kasutajateks on haiglad. Süsteem aitab lihtsustada kirurgia ja patoloogia valdkonna töötajate igapäevaseid protsesse. Näiteks tarkvara abil saab operatsiooni ajal kaugjuhida seadmeid ruumis või teha lihtsalt proovist pilte patoloogilise uuringu ajal. Praeguste kasutajate tagasiside põhjal soovitakse lisafunktsiooni ja nähtakse vajadust lisada aken, kus oleks üldinformatsioon patsiendi protseduuri kohta kuvatud kasutaja jaoks mugavaima disainilise lahendusega. Hetkel on informatsioon kättesaadav iga funktsiooni enda aknas või rakenduses peidetult.

Lõputöö peamiseks eesmärgiks on meditsiinilise tarkvara *Study info* akna teisaldamine UWP-le ning et lahendus oleks sobilik ettevõtte uue tootega. Teiseks eesmärgiks on võrrelda WPF (*Windows Presentation Foundation*) ja UWP lahendust kirjanduse ja enda tehtud koodi põhjal ja analüüsida nende eelised ja puudused. Kolmandaks eesmärgiks

koostada WPF-st UWP-le teisaldamise juhend, mis oleks abiks teistele, kes plaanivad uuele platvormile üle minna.

1.2 Lühiülevaade teostatud funktsionaalsustest

Projekti käigus on teostatud kõik nõuded, mis ettevõtte projektile seadis. Teostati põhiline funktsionaalsus kuvada patoloogilise uuringu informatsioon UWP platvormil. Selle teostamiseks täideti lõputöös märgitud funktsionaalsed ja mittefunktsionaalsed nõuded, mille hulka käisid järgnevad punktid.

- Kasutada Visual Studiot ja UWP platvormi
- Kasutada disainiks MVVM (*Model-View-ViewModel*) arhitektuuri
- Kasutada koodi testimiseks ühikteste ja AAA (*Arrange, Act, Assert*) süntaksit
- Kasutada etteantud andmebaasi
- Kasutada etteantud projekti struktuuri
- Kasutajaliides oleks lihtsa ülesehitusega, kasutajasõbralik ning sarnane päris rakenduse osadega
- Patsiendi andmete kuvamine
- Konteineritega seotud andmete kuvamine
- Kassettidega seotud andmete kuvamine
- Uuringuga seotud kommentaaride kuvamine, muutmine ja salvestamine
- Meedia andmete kuvamine

Ettevõtte jaoks oli oluline akna lihtne kasutatavus ning selle tõttu ka *Study info* kasutajaliidese parandamine, mida teostati projekti lõpufaasis. Oluliseks lisandiks eelnevale projektile olid veel andmebaasi ja süsteemi vajalike paranduste teostamine. Kõik projektile seatud nõuded said täidetud ja funktsionaalsused teostatud.

1.3 Töö edasine struktuur

Käesolev töö koosneb viiest peatükist. Teises peatükis on antud täpne kirjeldus projektist, välja toodud tööriistad, tööprotsess. Kolmandas peatükis tehniline dokumentatsioon, kus on ülevaade projekti nõuetest, arhitektuurist, disainist, koodist ja testidest. Neljandas peatükis on tehtud analüüs, järeldused, põhjendatakse töö tulemusi, on välja toodud tööde logid ja hinnatakse ühiselt meeskonnatööd. Lisaks on peatükis kirjalik lõputöö osa UWP

ja WPF võrdluse kohta ning juhend teisaldamiseks. Viimaseks peatükiks on lõputöö kokkuvõte.

2 Metoodika

2.1 Objekti detailne kirjeldus

Study info loomise peamine põhjus on lihtsustada meditsiinitarkvara rakenduses kasutajate tööd. Üldinfo kuvamine ühes aknas võimaldab leida kõik samast kohast ning teha protseduurid seotult patoloogilise töö lõpetamisega ja/või analüüsimisega. Varasemalt selline võimalus puudus ja uuringu kohta kokkuvõtvat teavet kasutaja ei saanud. Ettevõtte soovist tõsta kogu rakendus üle UWP-le tekkis ka vajadus *Study info* aknasse teha muudatusi ja panna sobituma UWP platvormiga. Tarkvara sooviti üle viia platvormile, sellepärast et seal on uuemad UI (*User Interface*) komponendid. Lisaks on UWP-s sisseehitatud tarkvara automaatne uuendamine ja loomulik puuetundlikkus. See on oluline ja vajalik, kuna tarkvara kasutatakse puuetundlikul ekraanil ja see hõlbustab programmi kasutamist.

Study info akna arendamisega on tekkinud väljavaade kogu informatsioonist, mida laboritöötajal on tarvis uuringu lõpetamiseks, nt konteinerite või kassetide ülevaade. Lisaks üldinfole, mis hõlmavad peamiselt patsiendi, uuringu ja selle teostajaga seotud teavet on võimalus näha ka meediaga seonduvat informatsiooni, mis kuvab loetelu selle sisust, mitu faili on salvestatud, mis formaadis ja mitu pilti on saadetud piltide arhiveerimise ja kommunikatsiooni süsteemi, mis võimaldab ökonoomset salvestamist ja mugavat juurdepääsu piltidele. Aken jaguneb järgnevalt uuringu üldinfo (Tabel 1) ja patoloogia uuringuga seonduvaks infoks (Tabel 2), mille alla käib proovi info (Tabel 3).

Tabel 1. Uuringu üldinformatsiooni väljad.

Uuringu üldinformatsioon
Algusaeg (<i>Start time</i>)
Lõppaeg (<i>End time</i>)
Uuringu id (<i>Study id</i>)
Protseduuri nimi (<i>Procedure name</i>)
Protseduuri kood (<i>Procedure code</i>)

Uuringu üldinformatsioon
Läbiviija (<i>Performer</i>)
Assistent (<i>Assistent</i>)
Patsiendi nimi (<i>Patient name</i>)
Patsiendi SSN (<i>Patient Social Security Number</i>)
Kommentaar (<i>Comment</i>)
Piltide hulk (<i>List of recorded images</i>)
Videote hulk (<i>List of recorded videos</i>)
Audiofailide hulk (<i>List of recorded audio</i>)
Arhiivi saadetud pildid (<i>Files sent to Picture archiving and communication system, No. and time</i>)

Tabel 2. Patoloogia uuringu väljad.

Patoloogia uuringu informatsioon
Uuringu tähtsus kategooria (<i>Study category level</i>)
Ohtliku sisu hoiatus (<i>Contamination warning</i>)
Nõelte oht (<i>Warning for needles</i>)
Konteinerite kogus (<i>Total quantity of containers</i>)
Konteineritega seotud kassetid (<i>Related cassettes for each containers</i>)
Kassetide kogus (<i>Total quantity of cassettes</i>)
Kassetide suurus (<i>Size of cassettes</i>)
Kassetis olevate proovide hulk (<i>No. of samples in 1 cassette</i>)
Lisatud kassetide kogus (<i>Quantity of added cassettes</i>)
Tellitud ja kasutatud kassetide kogus (<i>Quantity of ordered and used cassettes</i>)

Tabel 3. Patoloogia uuringuga seotud proovi väljad.

Proovi informatsioon
Makroskoopiline kirjeldus (<i>Macroscopic description</i>)
Esialgne informatsioon (<i>Preliminary information</i>)
Proovi suurus, kaal (<i>Sample size, weight</i>)

Study infol kuvatavatest andmetest vajasis andmebaasis parandust assistendiga seotud väljad. Arendamiseks anti kasutada Visual Studio projekt ja osa ettevõtte kasutatavast andmebaasist.

Projekti käigus lisandus kood äriloogikaga olemasolevatesse klassidesse *Examination*, *Examiner*, *ExaminationType*, *Patient*, *Container*, *ContainerType* ja nendega seotud repositooriumitesse. Vaatemudelid *StudyInfoViewModel*, *ContainerViewModel*, *CassetteViewModel* lisati olemasolevasse projekti *Facade* ning *Study Info* akna kasutajaliidese arendamiseks lisati juurde projekt koos vaatega *StudyInfoWindowView*.

2.2 Tööriistade kirjeldus

Projekti arenduskeskkonnaks oli Visual Studio, Microsoft SQL Server Management Studio ja peamiseks keelteks C#, XAML (*Extensible Application Markup Language*) ja SQL (*Structured Query Language*). Selleks et kiirendada programmi arendamist ja hooldatavust kasutati MVVM mustrit. Koodi refaktoormiseks oli kasutatud ReSharper. Versiooni kontrollimise tööriistaks kasutati GitLabi ja logimiseks Toggl keskkonda.

2.3 Tööprotsessi kirjeldus

Meeskonna siseselt otsustati, et projekti panustavad kõik liikmed võrdselt, mis tähendab et kõik kirjutavad koodi, teste ja arendavad vaadet. Arendusmetoodikas kasutati ka paarisprogrammeerimist, kuna nii saadi kõige paremini üksteist toetada tööprotsessi käigus ja areneda. Antud arendusmetoodika aitas kiirendada probleemide lahendamist ja vältis segadust tiimitöös. Töö jooksul püüti järgida agiilse arendamise printsiipe. Meeskonna kohtumised toimusid kord nädalas. Kokkusaamised toimusid koolis ja MS Teamsi vahendusel. Juhendajatega kohtuti vastavalt vajadusele MS Teamsi keskkonnas, kui oli vaja anda täiendavaid juhiseid või ülevaadet töö kulgemisest. Juhendajatega kokkusaamine toimus vähemalt kord kahe nädala tagant.

Tööprotsess algas uue platvormi analüüsimisega, õppimise ja tööprotsessi planeerimisega. Leiti, et mõned ülesanded saab omavahel ära jaotada, kuna aken on jaotatud blokkideks. Sellise korralduse juures suhtlesid meeskonnaliikmed aga pidevalt ja jagasid oma mõtteid töö edenemise kohta. Nii sai rasketes kohtades teisele abiks olla ja oldi teadlik, kuidas projekt areneb. Mõned ülesandegaotused kujunesid vastavalt projekti käigus ümber. Peamiselt oli ülesanded jaotatud tiimiliikmete vahel järgnevalt. Esimene tiimiliige tegeles rohkem patsiendi informatsiooni, uuringu hoiatuste, meedia ja testimisega. Teine tiimiliige uuringu informatsiooni, patoloogia informatsiooni ning

kasutajaliidese disainiga. Koostöös valmis akna keskmine osa, mis on seotud patoloogia konteinerite ja kassettidega, olles ka üks keerulisemaid akna osasid.

3 Tulemused

Projekti arendati Bait Partner OÜ poolt meeskonnale väljastatud nõuete abil, mis on välja toodud järgnevas peatükis, koos tehnilise dokumentatsiooniga. *Study info* aknale kuvati töö tulemusena planeeritud infoblokid ning aken viidi UWP raamistikku. Aken arendati eraldiseisva blokina ning on seotav olemasoleva ettevõtte infosüsteemiga ja sobib ettevõtte meditsiinitarkvara külge.

3.1 Süsteemi funktsionaalsed ja mittefunktsionaalsed nõuded

Ettevõtte poolt on antud järgmised mittefunktsionaalsed nõuded:

- Kasutada Visual Studiot ja UWP platvormi
- Kasutada süsteemi disainiks MVVM arhitektuuri
- Kasutada koodi testimiseks ühiktestide ja *Arrange, Act, Assert (AAA)* süntaksit
- Kasutada etteantud andmebaasi
- Kasutada etteantud projekti struktuuri
- Kasutajaliides oleks lihtsa ülesehitusega, kasutajasõbralik ning sarnane päris rakenduse osadega

Funktsionaalseteks nõueteks olid:

- Patsiendi andmete kuvamine
- Konteineritega seotud andmete kuvamine
- Kassettidega seotud andmete kuvamine
- Uuringuga seotud kommentaaride kuvamine, muutmine ja salvestamine
- Meedia andmete kuvamine

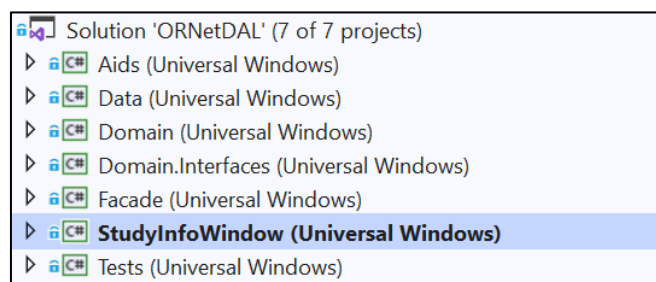
3.2 Arhitektuur

Projekti arhitektuur

Projekti arendamisel anti kasutada mitmekihilise arhitektuuriga koodi. Kogu rakendus jagunes seitsmeks projektiks (Joonis 1). Projekti arendamisel kasutati põhiliselt

äriloogika (*Domain*), andmekihi (*Data*), esitluskihi (*Facade*, *StudyInfoWindow*) ja test (*Test*) projekte. Rakenduse on jagatud projektideks järgnevalt.

- *Aids*– projekt, milles on rakenduse jaoks vajalikud abiklassid
- *Data*– projekt, milles asuvad repositooriumid ja meetodid andmebaasiga suhtluseks
- *Domain*– projekt, milles asuvad objektide mudelid
- *Domain.Interfaces*– projekt, milles asuvad vajalikud abiklassid põhiliselt *Domain* projektile
- *Facade*– projekt, milles asuvad vaadete mudelid
- *StudyInfoWindow*– projekt, kus asub *Study info* vaade
- *Tests*– projekt, milles asuvad vaatemudelite *StudyInfoViewModel*, *ContainerViewModel*, *CassetteViewModel* ühiktestid



Joonis 1. Projekti arhitektuur.

Andmebaasi arhitektuur

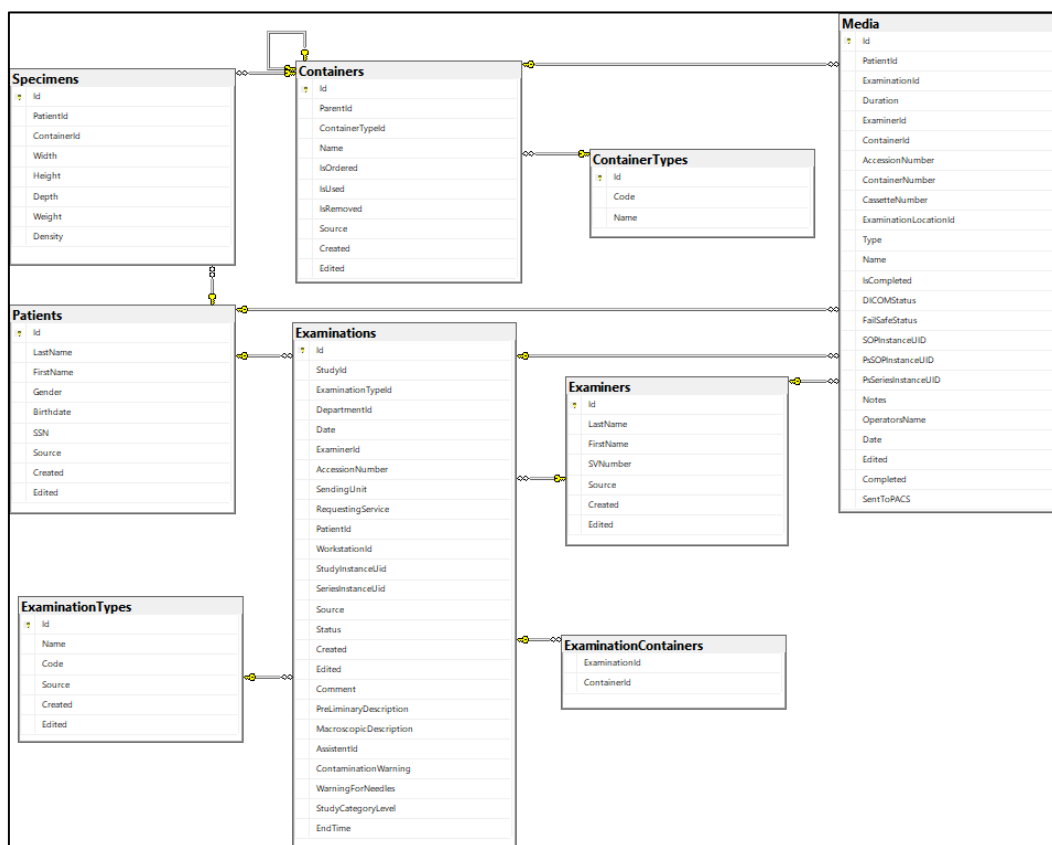
Andmebaas oli ettevõtte poolt antud osa nende kasutatavast andmebaasist. Sellest suurest andmebaasist omakorda kasutati projekti jaoks järgnevat üheksat tabelit.

- **Patsiendid (*Patients*)**– patsiendiga seonduvad read, tabel on seotud proovid, meedia ja uuringud tabelitega
- **Uuringud (*Examinations*)**– uuringu andmetega seonduvad read, tabel on seotud patsiendid, meedia, läbiviija, uuringutüübid ja uuringu konteinerid tabelitega
- **Uuringutüübid (*ExaminationTypes*)**– uuringutüübiga seonduvad read, tabel on seotud uuringute tabeliga
- **Uuringu konteinerid (*ExaminationContainers*)**– uuringu konteineritega seonduvad read, tabel on seotud uuringute tabeliga
- **Läbiviijad (*Examiners*)**– uuringu läbiviijatega seonduvad read, tabel on seotud meedia ja uuringute tabelitega

- **Meedia (*Media*)**– meediaga seonduvad read, tabel on seotud konteinerite, patsientide, uuringute ja läbivijjate tabelitega
- **Proovid (*Specimens*)**– prooviga seonduvad read, tabel on seotud konteinerite ja patsientide tabelitega
- **Konteinerid (*Containers*)**– konteineriga seonduvad read, tabel on seotud meedia, proovide ja konteinerite tabelitega
- **Konteineri tüübid (*ContainerTypes*)**– konteineri tüübiga seonduvad read, tabel on seotud konteinerite tabeliga

Andmebaasi tabelid seotud assistendiga vajasis projekti käigus parandamist. Ülejäänud väljad olid varasemalt juba olemas.

Järgnevalt on näha tabelid suurest andmebaasist, mida akna arendamisel vajati ning nende omavahelised seosed (Joonis 2).



Joonis 2. Andmebaasi arhitektuur.

Study Info aknal kuvatakse ühe uuringu informatsioon, sellepärast tuginetakse seoste selgitamisel konkreetsele uuringule. Andmebaasi tabelist saab välja lugeda, et uuringul saab olla üks läbivijja, üks uuringu tüüp ja üks patsient. Uuringul on tabelitega *Media* ja

ExaminationContainers üks mitmele seos, mis defineerib, et uuringul saab olla mitu meediafaili ja mitu konteinerit. Uuringul on üks patsient ja temaga on seotud üks mitmele suhtes *Media* ja *Specimens* tabelid. Patsiendil saab olla mitu proovi ning seotud meediafaili. *Study Info* aknal kuvatav meedia on seotud nii patsiendi kui ka uuringuga.

3.3 Disain

Süsteemi disain

Projekti arendamisel kasutati MVVM disainimustrit, mis oli ettevõtte poolt soovitud lähenemine. MVVM muster jaguneb mudeliks (*Model*), vaateks (*View*) ja vaatemudeliks (*ViewModel*). Mudeli ülesandeks on tagada erinevatele andmeallikatele ligipääs. Vaade esitab andmeid visuaalselt, näitab andmete olekuid ja suhtleb kasutajaga. Mudel ja vaade hoitakse võimalikult lihtsad ning koodi põhiosa on vaatemudelis. Vaatemudel on siduv osa disainimustril ning edastab suhtluse mudeli ning vaate vahel. Projekti näitel asuvad mudelid projektis *Domain*, vaade projektis *StudyInfoWindow* ja vaatemudelid projektis *Facade* [2].

Kasutajaliidese disain

Kasutajaliidese disainil toetuti suuresti kasutajasõbralikkusele ja varasemalt meeskonnaprojekti tarvis loodud akna disainile ning selle täiustamisele. Kuna kasutajateks on meditsiinitöötajad on disaini lihtsus oluline. Olemasolevas rakenduses kasutatakse palju ikoone, et lihtsalt informatsioon kasutajale selgeks teha, nii kasutati neid ka *Study info* aknas. Kõik ikoonid ja logod on ettevõtte poolt kasutuses ja antud aknal kasutamiseks. Akna arendamisel lähtuti olemasoleva rakenduse disainist ja ka uuendustest, mida sooviti UWP platvormil disaini poolest näha. Disainimuudatuste jaoks konsulteeriti ettevõttega ja saadi juhiseid ettevõtte poolsetelt juhendajatelt. Kasutajaliides on jaotatud kolmeks blokiks (patsiendi, patoloogia, meedia andmed), kus on eritüüpi informatsioon eraldi. *Study info* akna vaade on toodud lisas 4.

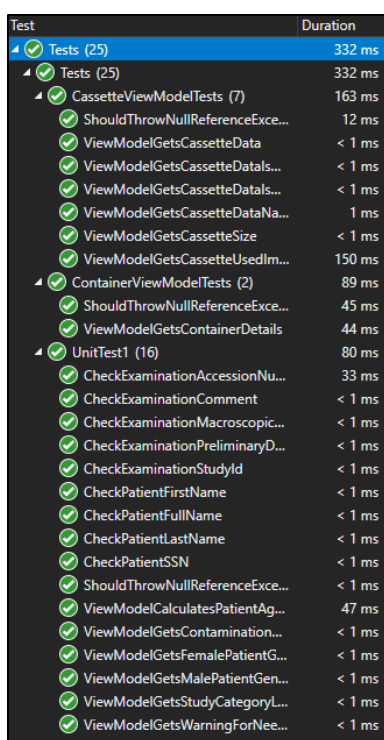
3.4 Kood

Arendamiseks kasutati *backend*-is C# programmeerimiskeelt. Andmebaasi andmete kasutamiseks rakendati SQL andmebaasi päringukeelt. *Study info* akna kasutajaliidese elementide määratlemiseks ja sidumiseks andmetega kasutati XAML märgituskeelt.

Koodi kirjutamisel jälgiti puhta koodi ideoloogiad– nimed on selged, eesmärgipärased ja ühekeelsed, meetodid on lühikesed ja täidavad ühte või väheseid ülesandeid, kood on arusaadav ning ei vaja kommentaare.

3.5 Testid

Projekti käigus testiti ühiktestidega vaatemudelite klasse. Ühiktestide tegemine oli oluline selleks, et olla kindel kuidas meetodid toimivad. Testimisel kasutati AAA süntaksit, mis aitab hoida testmeetodite struktuuri. Sisendväärtusteks olid testandmed ja nullväärtused. Joonisel 3 on toodud läbitud ühiktestid.



Test	Duration
Tests (25)	332 ms
Tests (25)	332 ms
CassetteViewModelTests (7)	163 ms
ShouldThrowNullReferenceExce...	12 ms
ViewModelGetsCassetteData	< 1 ms
ViewModelGetsCassetteData...	< 1 ms
ViewModelGetsCassetteData...	< 1 ms
ViewModelGetsCassetteDataNa...	1 ms
ViewModelGetsCassetteSize	< 1 ms
ViewModelGetsCassetteUsedIm...	150 ms
ContainerViewModelTests (2)	89 ms
ShouldThrowNullReferenceExce...	45 ms
ViewModelGetsContainerDetails	44 ms
UnitTest1 (16)	80 ms
CheckExaminationAccessionNu...	33 ms
CheckExaminationComment	< 1 ms
CheckExaminationMacroscopic...	< 1 ms
CheckExaminationPreliminaryD...	< 1 ms
CheckExaminationStudyId	< 1 ms
CheckPatientFirstName	< 1 ms
CheckPatientFullName	< 1 ms
CheckPatientLastName	< 1 ms
CheckPatientSSN	< 1 ms
ShouldThrowNullReferenceExce...	< 1 ms
ViewModelCalculatesPatientAg...	47 ms
ViewModelGetsContamination...	< 1 ms
ViewModelGetsFemalePatientG...	< 1 ms
ViewModelGetsMalePatientGen...	< 1 ms
ViewModelGetsStudyCategoryL...	< 1 ms
ViewModelGetsWarningForNee...	< 1 ms

Joonis 3. Läbitud testid.

Lisaks sellele oli aken testitud manuaalselt. Erinevate stsenaariumite jaoks kasutati andmeid andmebaasist, et veenduda, et kõik kuvatakse õigesti ja vaadati, kuidas aken toimib erinevates olukordades.

4 Analüüs

Antud peatükis analüüsitakse tehtud tööd. Tuuakse välja lõputöö analüüs WPF ja UWP võrdlemisel ja juhend *Universal Windows* platvormile üleminekuks.

4.1 Tehnilise teostuse analüüs

Nõuded

Projekti nõuded olid ettevõtte poolt määratletud ning teostati vastavalt. Nõuded platvormi ja arenduskeskkonna kasutusele on tulnud ettevõtte kasutatavatest tarkvaradest ning tehnoloogiatest. Kasutati näiteks Visual Studiot ja Microsoft SQL Server Management Studiot, sest mõlemad on UWP rakenduste arendamiseks sobilikud. MVVM arhitektuuri kasutus hõlpsustab koodi osade muutmist ilma suurte muudatuste tegemiseta terves koodis. Testimine oli oluline, et kontrollida koodi õiget toimimist.

Arhitektuur

Projektide arhitektuur oli mitmekihiline, et selgelt eraldada erinevad projekti osad. Selline grupeerimine võimaldab koodi teha programmeerijale loetavamaks ja arendamisel hästi jälgida puhta koodi printsiipe. Teisisõnu, selline arhitektuur on hea koodi osade muutmiseks tulevikus ning koodi taaskasutamiseks tulevates projektides.

Disain

Projektis disainimustrina kasutati MVVM arhitektuurimustrit. Mudelite, vaatemudelite ja vaate jagamine võimaldab eraldada kasutajaliidese rakenduse loogikast. Selline lähenemine tagab parema rakenduse hooldatavuse ja osade eraldatuse, mis võimaldas ka tiimitöös ülesandeid paremini jagada. Vaadete eraldamine loogikast võimaldab selle korduvkasutatavust väheste muudatustega. MVVM arhitektuur on ka kasuks ühiktestimisel.

Kasutajaliidese disaini tingimused olid suuresti ettevõtte poolt määratletud ning uuendused läbiräägitud.

Kood

Koodi kirjutamisel kasutati üldtuntud puhta koodi printsiipe. Projekti vältel kasutati ühtlast stiili ja peeti meeles koodi loetavust tulevikus. Koodi kirjutamisel oli kasutuses ReSharper, mis aitas kaasa koodi struktuuri vigade tuvastamisel.

Kasutajaliideseid saab deklaratiivselt kirjeldada XAML failides. Neid hoitakse sõltumata *code-behind*-ist mis aitab eraldada välimust funktsionaalsusest [3]. See teeb paralleelset arendamist lihtsamaks, kuna arendajad saavad tegeleda oma osadega [4]. Niimoodi saab mugavalt jagada ülesanded, kus näiteks üks arendaja tegeleb puhtalt UI-ga samal ajal, kui teine kirjutab *back-end*-i jaoks koodi.

Lõputöös kirjutatud koodiridu on ligikaudu 4000.

Testid

Testimiseks kasutati MSTest raamistikku ja kirjutati ühikteste vaatemudelite jaoks. Ettevõtte poolt oli soovitatud testimise jaoks kasutada NUnit raamistikku. Esialgu sooviti kasutada *Unit Test Project*-i, mis kasutab NUnit-it, millega autorid olid ka enne kokku puutunud. Seda kasutades tekkis aga konflikt, sest UWP projektid ei läinud kokku .NET Core projektiga. Prooviti teha *Class Library*, selleks et lisada sellesse klasse, mida on vaja testida, kuid ka see ei toiminud. Soovitused netiallikatest on kasutada UWP jaoks spetsiaalset *Unit Test App*-i (*Universal Windows*) [5]. Kuna selle kohta on kõige rohkem informatsiooni ja see on töökindel variant, siis lõpuks otsustatigi kasutada seda.

Töö käigus selgus ka selle puudus. Vaatamata sellele, et seda soovitakse juba mitu aastat, ei ole koodi katvus toetatud UWP ühiktestide jaoks. Alles hiljuti kirjutas Microsoft avatud lähtekoodiga projektist, mida saab kasutada selleks, et vaadata mitu protsendi koodist on kaetud testidega [6]. Kuna projekti käigus ei jõutud kirjutada piisavalt palju teste, siis otsustati mitte proovida avatud lähtekoodiga projekti ja pigem keskenduda põhieesmärkidele.

4.2 Kirjanduse ülevaade

UWP ja WPF sarnasused ja erinevused.

Ajalugu

WPF ja UWP kasutajaliidesed on mõlemad vektoripõhised. Mõlemad raamistikud on Windowsi töölaua rakenduste arendamiseks. WPF on UWP eelkäia, mis avaldati esmalt 2006 aasta. WPF'i saab kasutada Windowsi versioonidega alates XP'st. WPF oli esimesi moodsaid lahendusi alternatiiviks Windows Forms või Win32-le. Seda kasutatakse laialdaselt ka tänapäeval [7].

UWP on platvorm, mis on enam kui vaid graafiline kasutajaliidese raamistik. See sobib kasutamiseks Windows 10-ga ning selle uuenduslikuks küljeks varasematele lahendustele on võimalus kirjutada jagatud koodi asemel eri Windowsi seadmetele üks kood, mis sobib kõigile neile kasutamiseks [8].

Operatsioonisüsteemid

UWP rakendust saab kasutada kõigi Windows 10 versioonidega. WPF toimib alates versioonist Windows 7. WPF toimib ka Windows XP-l, kui on laetud alla vastav .NET raamistik [9].

Keeled

Mõlemate levinuimateks arenduskeelteks on C#, C++, VB, XAML. UWP võimaldab kasutada veel Javascripti ja HTML-i (*HyperText Markup Language*). WPF-is saab kasutada lisaks F# keelt [10].

Raamistiku

UWP kasutab Windows 10 SDK-d (*Software development kit*) ja sobib .NET Standardit rakendavate raamistikega. WPF on osa .NET või .NET Framework raamistikust [10].

Kasutajaliidese käitusaeg

Käitusel on UWP rakendust võimalik kompileerida *native* või *managed* koodina, WPF rakendust vaid *managed* koodina. *Native* koodi ja *managed* koodi vahe seisneb koodi käivitamises. *Native* kood ei käivitu tööajal vaid otse masina protsessori poolt ja on konfigureeritud töötama kindlal protsessoril. Kood ei ole mõeldud töötamiseks teistel protsessoritel ja töötab optimaalselt vaid määratud protsessoriga. UWP rakenduses tähendab see, et rakendus käivitub otse operatsioonisüsteemilt. *Managed* koodi

kasutamine UWP-s toimub läbi eelkompileerimise tehnoloogia .Net Native. See kompileerib automaatselt *managed* koodiga kirjutatud rakendused, mis kasutavad .NET Frameworki ja Windows 10-t, *native* koodi [11].

Managed kood aga on kood, mille täitmist hallatakse käivitamisel. Kood on algselt kirjutatud mistahes kõrgetasemelise programmeerimiskeele abil ja toetatud .Net Frameworki või .NET Core'i raamistike poolt. Käitusaja esimeses pooles tõlgitakse kõrgetasemeline kood vahekeeleks (*Intermediate Language*). Vahekeelt hoitakse eraldi failis koos metaandmetega, mis kirjeldavad loodud koodi atribuute. Käitusaja teisel poolel kirjutatakse vahekeel masinkoodiks, mis sobib konkreetse masinaga, millel kood käivitatakse [12].

XAML

XAML on deklaratiivne keel, mis põhineb XML-il (*Extensible Markup Language*). XAML-i kasutatakse kasutajaliideste loomiseks sellistes rakendustes, nagu WPF, UWP ja Xamarin.Forms [13].

Koodi teisendamine võib tekitada raskusi, kuna XAML-il on erinevad dialektid [14]. Näiteks *multibinding*, mis võimaldab siduda korraga mitu väärtust, on funktsioon, mis esineb ainult WPF-s. Alternatiiv sellele on kasutada *Run* elemendi. Allpool on toodud välja autorite WPF ja UWP rakenduste koodilõigud (Joonis 4-9).

```
<TextBlock Grid.Row="0" Grid.Column="0" Margin="8">
  <Run Text="Container"/>
  <Run Text=" "/>
  <Run Text="{Binding Name}"/>
</TextBlock>
```

Joonis 4. UWP rakenduse XAML *Run* element.

```
<TextBlock>
  <TextBlock.Text>
    <MultiBinding StringFormat="Container {0}">
      <Binding Path="Name" />
    </MultiBinding>
  </TextBlock.Text>
</TextBlock>
```

Joonis 5. WPF rakenduse XAML *multibinding*.

Nimeruumide (*namespaces*) importimine XAML dialektides erineb. WPF rakendusel XML nimeruum „`http://schemas.microsoft.com/winfx/2006/xaml/presentation`“ kujutab .NET nimeruumi *System.Windows.Controls*. UWP puhul on see WinRT nimeruum *Windows.UI.Xaml.Controls* [15]. Nimeruumide importimiseks WPF rakenduses kasutatakse *clr-namespace* süntaksit, UWP rakenduses *using* süntaksit. Näidetena on toodud autorite poolt tehtud WPF ja UWP projektide vaadete (*Views*) koodiread.

```
xmlns:viewModels1="clr-namespace:WpfApp1.ViewModels"
```

Joonis 6. WPF rakenduse XAML nimeruumide importimine.

```
xmlns:controls="using:Microsoft.Toolkit.Uwp.UI.Controls"  
xmlns:viewmodels="using:StudyInfoWindow.ViewModels"
```

Joonis 7. UWP rakenduse XAML nimeruumide importimine.

DockPanel on üks klassidest, mida UWP rakendus ei toeta. Seda kasutatakse selleks, et määratleda ala, kus asuvad alamelemendid vertikaalselt või horisontaalselt üksteise vastu [16]. UWP rakenduses seda saab määratleda, kasutades *Community Toolkit*-i. *Windows Community Toolkit* on tööriistakomplekt, mis koosneb erinevatest abifunktsioonidest, rakenduste teenustest ja kohandatud kontrollidest (*custom controls*) [17].

```
<DataTemplate>  
  <controls:DockPanel Width="100">  
    <TextBlock>  
      <Run Text="{Binding Name}"/>  
      <Run Text=" "/>  
      <Run Text="{Binding Size}"/>  
    </TextBlock>  
  </controls:DockPanel>  
</DataTemplate>
```

Joonis 8. UWP rakenduse DockPanel näide.

```
<DataTemplate>  
  <DockPanel Width="400">  
    <TextBlock Text="Cassette nr: " />  
    <TextBlock FontWeight="Bold" Text="{Binding Name}" />  
    <TextBlock Text=" Cassete size: " />  
    <TextBlock Text="{Binding Size}" FontWeight="Bold" />  
  </DockPanel>  
</DataTemplate>
```

Joonis 9. WPF rakenduse DockPanel näide.

Ühiktestid

UWP ja WPF rakenduse ühiktestimiseks on kasutuses MSTest ja XUnit raamistikud. WPF kasutab veel NUnitit, UWP *library*-te testimiseks ei ole NUnit hästi toetatud. Kasutades UWP projektis .Net Standard *library*-it saab NUnit teste kasutada.

Juurutamine

UWP rakenduse saab avaldada mitmel viisil. Esimeseks võimaluseks on Microsoft Store, mille kaudu on rakendus allalaaditav ja avalik kõikidele. Selleks on vajalik ka *Windows Developer License*. Teiseks võimaluseks on avaldada rakendus Microsoft Store for Businessis, mis on ettevõtetele mõeldud rakenduste pood, kus saab suunata, kellele on rakendus mõeldud. Need kaks võimalust muudavad UWP rakenduste avaldamise palju mugavamaks, kui WPF rakenduse [18].

Rakendusi saab juurutada ka ilma Microsoft Storeita. Selleks tuleb kasutada seadmehaldustööriistu, konfigureerida rakenduse installimisprogrammil põhinevat juurutamist ja rakendusi otse laadida. Selleks on vajalik omada sertifikaati ja allalaetavas seadmes lubada arendusrežiim [18].

WPF rakenduse juurutamiseks kasutatav tehnoloogia sõltub suuresti rakenduse tüübist. Eraldi kasutatavaid rakendusi juurutatakse kas ClickOnce või Windows Installeriga. Nad vajavad luba, mis Windows Installeri puhul on automaatne [19].

Markup-Only XAML rakendused on veebiserveritesse juurutatud lehed, mida saab kasutada veebibrauseris Internet Explorer või Windows Explorer. Sellised rakendused saab juurutada kasutades XCopy või Windows Installeri tehnoloogiat [19].

XAML brauseri rakendused vajavad juurutamiseks kolme faili ja neid juurutatakse sarnaselt veebiserverisse ja on võimalik juurutada nii ClickOnce, Windows Installeri kui ka XCopy abil [19].

Kiirus

Native kood on üldiselt kiirem kui *managed* kood, mida kasutatakse WPF rakendustes. *Managed* koodi puhul vahekoodi ei saa käivitada otse läbi CPU (*Central Processing Unit*). Programmi poolt rakenduse laadimisel teisendatakse vahekood tegeliku protsessori

native koodiks. *Native* koodiks tõlkimine võib tekitada viivitusi käivitamise ajal. *Managed* kood nõuab rohkem mälu. UWP rakendused, mis on kirjutatud C# või VB keeltes kompileeritakse *.NET Native*’iga, mis teeb nende soorituse sama kõrgeks, kui *native* rakendusel [20].

Selleks, et kontrollida teorias saadud teadmised kiiruse kohta on läbi viidud analüüs eelmise semestri meeskonna- ja lõputöö raames tehtud rakenduste põhjal. WPF rakenduse puhul on kommenteeritud välja kood, mis ei ole kasutuses UWP rakenduses, selleks et rakendused ja nende funktsionaalsused oleksid identsed. Joonistel 10 ja 11 37,6 MB (*Megabyte*) WPF puhul ja 29,6 MB UWP puhul näitavad akende mälu kasutamist peale nende avamist. Antud korral midagi akendel ei avatud ega puudutatud. Nagu on näha, WPF rakendus võtab rohkem mälu.

Järgmisena on proovitud kasutada konteinerite nimekirja kerimisriba. Joonistel 12 ja 13 protsendid 4,3% ja 2,4% näitavad, kui palju WPF ja UWP kasutavad protsessori võimsusest antud hetkel. WPF rakendus kasutab kaks korda rohkem võimsust, mis on halb tulemus tavalise kerimise puhul. Lisaks sellele UWP rakenduse kerimine on sujuvam, mis on eriti tähtis kasutajamugavuse mõttes. Ühe uuringu konteinereid võib olla palju ja aeglane ning mittesujuv kerimine võib tekitada arstidele kasutajatena ebamugavust.

WpfApp1 (32 bit)	0%	37.6 MB	0 MB/s	0 Mbps
------------------	----	---------	--------	--------

Joonis 10. WPF rakenduse mälu kasutamine.

StudyInfoWindow (32 bit)	0.2%	29.6 MB	0 MB/s	0 Mbps
--------------------------	------	---------	--------	--------

Joonis 11. UWP rakenduse mälu kasutamine.

WpfApp1 (32 bit)	4.3%	49.9 MB	0 MB/s	0 Mbps
------------------	------	---------	--------	--------

Joonis 12. WPF rakenduse CPU kasutamine.

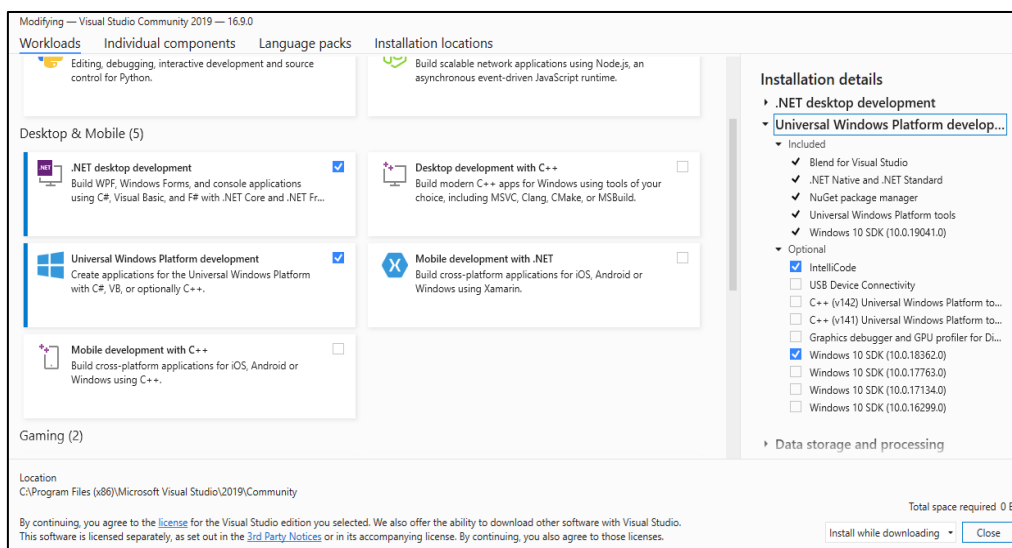
StudyInfoWindow (32 bit)	2.4%	26.2 MB	0 MB/s	0 Mbps
--------------------------	------	---------	--------	--------

Joonis 13. UWP rakenduse CPU kasutamine.

Rakenduse ületõstmise juhend WPF-st UWP platvormile

1. Arenduskeskkonna seadistamine

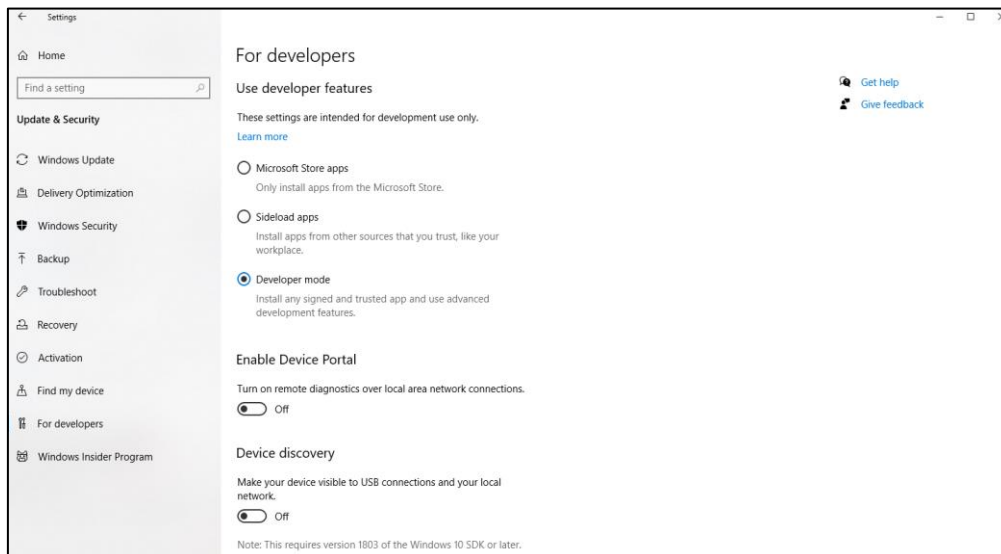
Selleks, et alustada UWP rakenduse arendamist, peab veenduma, et Visual Studio Installeris on *Desktop & Mobile* kategooria all valitud ja installitud *Universal Windows Platform development* (Joonis 14).



Joonis 14. Visual Studio Installeri installatsioon.

2. Arendusrežiimi lubamine

Järgmise sammuna on vaja valida sätete *Update & Security* alt aken “Arendajale” (*For developers*). Selle jaoks, et juurutada (*deploy*) ja debuggida (*debug*) tuleb valida *Developer mode* (Joonis 15). Peale seda võib tekkida vajadus arvuti taaskäivitada.



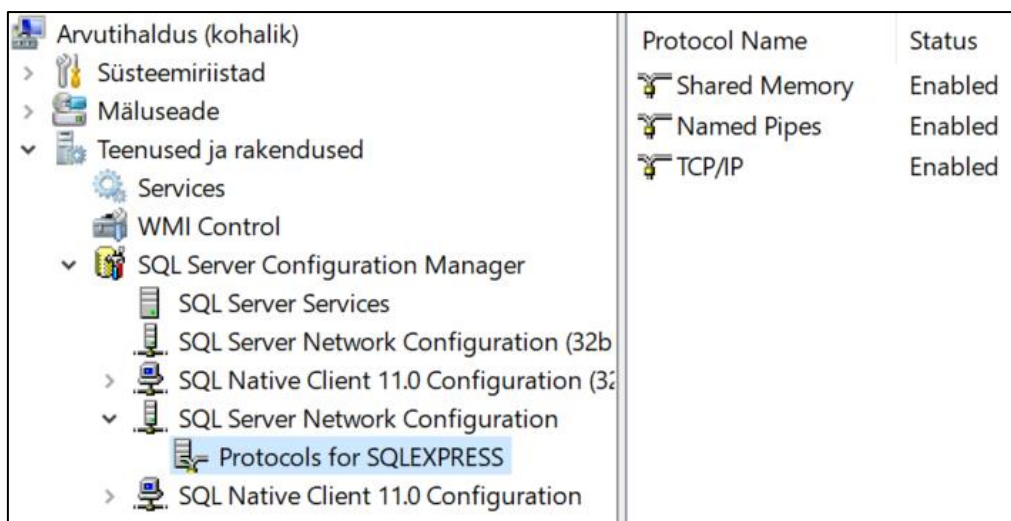
Joonis 15. Arendaja sätete lubamine.

3. Õige Windows 10 versiooni valimine sihtversiooniks

Valida tuleb kaks versiooni, siht (*target*) ja miinimum. Sihtversioon on Windows 10 versioon, mille peal plaanitakse rakendust jooksutada. Valides viimane võimalik versioon saadakse juurdepääs kõikidele hiljutistele API-dele (*Application Program Interface*). Miinimumversioon peab olema kas väiksem või võrdne sihtversiooniga. Valides erinevad versioonid peab arvestama sellega, et see tähendab, et rakendus töötab ükskõik mis versioonil selles vahemikus, mis omakorda tähendab lisatööd. Samasuguste versioonide puhul ei pea midagi tegema lisaks. See teeb tööd lihtsamaks ja aitab vältida võimalikud probleemid seotud versioonidega [21] [22].

4. Andmebaasi andmetele ligipääs

Andmete hoidmise jaoks saab kasutada SQLite, SQL Server, MongoDB või MySQL andmebaase. Enamik juhtudel tekib SQL Serveri andmebaasiga ühendamisel viga ning tuleb muuta „SQL Server“-i konfiguratsioone lubades „SQLEXPRESS“ protokollid „TCP/IP“ protokoll (Joonis 16) [23].



Joonis 16. TCP/IP protokollid lubamine.

5. .NET Native kasutamine

Rakendust on hea kontrollida *release* konfiguratsioonis .NET Native *toolchain*'iga, see aitab parandada ja optimeerida käituse jõudlust. UWP rakenduse avaldamisel Windows Store'is kompileeritakse see .NET Native'iga automaatselt. Seetõttu on hea testida rakendust juba varakult .NET Native *toolchain*-iga ja avastada *library*-id, mida UWP ei toeta [24].

6. Projektide ümbertõstmise õigele raamistikule

Kõik UWP rakenduse jaoks vajaminevad projektid tõstetakse üle kas *Universal Windows*-i või *.NET Standard* tüüpi projektiks. Protsessi käigus tulevad esile kohad, mis võivad olla erinevad varasema raamistiku kasutusest ja ei toeta uut raamistikku. Need tuleks muuta, leida alternatiivid ning UWP-s töötavad paketid [25].

7. ConnectionString

Tihti peale on andmebaasi ühenduseks vajaminev *connection string* defineeritud WPF rakenduses *App.config* failis. UWP rakendus sellist faili automaatselt kaasa ei tule ning kasutatakse andmebaasi ühenduse lisamist *code-behind* klassi *App.Connection string*-i hoidmine failis toimib lisades projekti *.resw*, *.xml* või *.json* tüüpi fail [23].

8. XAML muutuste parandamine

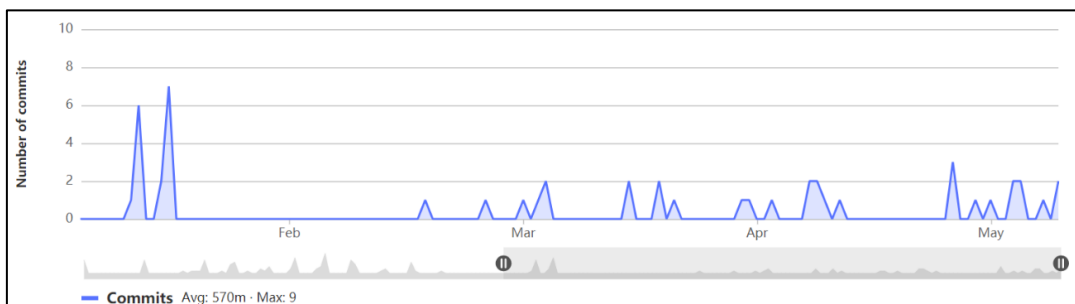
Kuna XAML-il on dialektid ning UWP ja WPF puhul võivad esineda erinevused, peab üle vaatama vaadete koodi ja seda vajadusel parandama.

9. Testimine

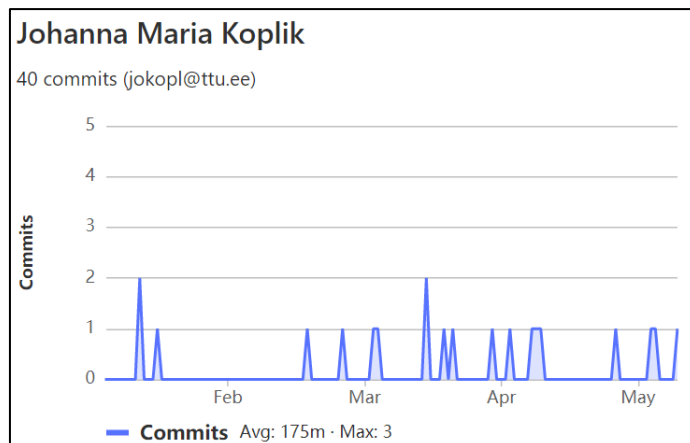
Ühiktestide tegemiseks tuleks kasutada spetsiaalset UWP ühiktestide projekti, milles kasutatakse MSTesti või muuta NuGet pakette vastavalt, et saaks kasutada XUnit teste.

4.3 Teostatud tööde detailne kirjeldus logi vormis

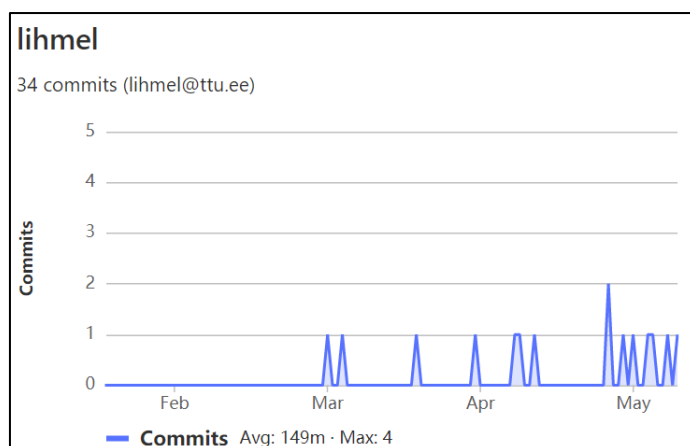
Väljavõte Gitlab commitidest :



Joonis 17. Commit-id projekti peale.



Joonis 18. *Commit*-id Johanna Maria Koplík.



Joonis 19. *Commit*-id Linda Hmelevskaja.

Väljavõte tegevustest ja ajakulu Toggl keskkonnas:

Johanna Maria Koplík



Joonis 20. Väljavõte ajakulust 1.

Date	Description	Duration	User
02/01	Projekti planeerimine BaitPartner	1:10:00 14:00-15:10	Johanna Koplik
02/12	Ettevõttega koosolek BaitPartner	1:15:00 14:30-15:45	Johanna Koplik
02/13	UWP-ga tutvumine BaitPartner	3:15:00 20:15-23:30	Johanna Koplik
02/19	UWP-ga tutvumine BaitPartner	2:50:00 14:30-17:20	Johanna Koplik
02/22	Projekti konfiguratsioonid BaitPartner	3:00:00 12:00-15:00	Johanna Koplik
02/24	Ülesandepüstitus BaitPartner	3:30:00 18:00-21:30	Johanna Koplik
03/01	Projekti konfiguratsioonid BaitPartner	4:00:00 12:00-16:00	Johanna Koplik
03/02	.NET standardi üleminek BaitPartner	6:43:27 09:46-16:30	Johanna Koplik
03/02	.NET standardi üleminek BaitPartner	3:40:00 15:30-19:10	Johanna Koplik
03/09	Projekti konfiguratsioonid BaitPartner	5:20:00 11:56-17:16	Johanna Koplik
03/10	Disain BaitPartner	4:10:00 13:30-17:40	Johanna Koplik
03/13	Tiimi koosolek BaitPartner	1:45:00 12:00-13:45	Johanna Koplik
03/15	Kirjandus BaitPartner	2:46:00 11:59-14:45	Johanna Koplik
03/18	Andmebaasi ühenduse parandamine ja binding BaitPartner	2:10:00 11:30-13:40	Johanna Koplik
03/19	Koosolek BaitPartner	1:30:00 12:00-13:30	Johanna Koplik

Joonis 21. Väljavõte ajakulust 2.

03/21	Andmebaasi ühenduse parandamine ja binding BaitPartner	5:45:00 12:30-18:15	Johanna Koplik
03/21	Andmebaasi ühenduse parandamine ja binding BaitPartner	1:15:00 19:30-20:45	Johanna Koplik
03/21	Parandused koodis BaitPartner	0:50:00 22:05-22:55	Johanna Koplik
03/24	Disain ja UWP õppimine BaitPartner	5:00:00 20:30-01:30	Johanna Koplik
03/25	Esimene blokk BaitPartner	6:10:00 14:50-21:00	Johanna Koplik
03/27	Kontainerite loogika BaitPartner	5:04:31 11:08-16:12	Johanna Koplik
03/28	Kontainerite loogika BaitPartner	2:33:42 18:56-21:30	Johanna Koplik
03/31	Koosolek ettevõttega BaitPartner	1:00:00 14:00-15:00	Johanna Koplik
03/31	Kirjandus BaitPartner	2:10:00 21:20-23:30	Johanna Koplik
04/03	Koosolek BaitPartner	1:30:00 14:00-15:30	Johanna Koplik
04/03	Esimene blokk BaitPartner	1:50:00 14:00-15:50	Johanna Koplik
04/04	Kontainerite loogika BaitPartner	1:00:00 14:50-15:50	Johanna Koplik
04/05	Tiimi koosolek BaitPartner	0:00:00 11:00-11:00	Johanna Koplik
04/05	Test projekt BaitPartner	1:50:00 11:00-12:50	Johanna Koplik
04/07	Kontainerite loogika BaitPartner	5:00:00 13:00-18:00	Johanna Koplik
04/09	Kontainerite loogika BaitPartner	5:00:00 14:30-19:30	Johanna Koplik
04/12	Test projekt BaitPartner	3:20:00 18:30-21:50	Johanna Koplik

Joonis 22. Väljavõte ajakulust 3.


04/14	Parandused BaitPartner	1:10:00 23:30-00:40	Johanna Koplik
04/15	Koosolek ja disain BaitPartner	3:40:00 14:20-18:00	Johanna Koplik
04/16	Kontainerite loogika BaitPartner	5:10:00 14:50-20:00	Johanna Koplik
04/16	Tiimi koosolek BaitPartner	1:00:00 22:00-23:00	Johanna Koplik
04/19	Tiimi koosolek BaitPartner	1:25:00 17:00-18:25	Johanna Koplik
04/21	Kirjandus BaitPartner	2:30:00 18:30-21:00	Johanna Koplik
04/22	Kirjandus BaitPartner	6:00:00 10:00-16:00	Johanna Koplik
04/22	Kirjandus BaitPartner	3:00:00 19:00-22:00	Johanna Koplik
04/24	Kirjandus BaitPartner	9:00:00 11:00-20:00	Johanna Koplik
04/25	Kirjandus BaitPartner	4:30:00 14:00-18:30	Johanna Koplik
04/26	Kirjandus BaitPartner	3:15:00 15:15-18:30	Johanna Koplik
04/26	Tiimi koosolek BaitPartner	1:00:00 17:00-18:00	Johanna Koplik
04/27	Kirjandus BaitPartner	5:30:00 17:00-22:30	Johanna Koplik
04/28	Disain BaitPartner	3:30:00 11:15-14:45	Johanna Koplik
04/28	Kirjandus BaitPartner	3:00:00 11:30-14:30	Johanna Koplik
04/28	Kirjandus BaitPartner	3:15:00 20:30-23:45	Johanna Koplik
05/02	Disain BaitPartner	2:40:00 10:00-12:40	Johanna Koplik
05/02	Disain BaitPartner	3:15:00 19:00-22:15	Johanna Koplik

Joonis 23. Väljavõte ajakulust 4.

05/03	Disain BaitPartner	4:30:00 11.30-16.00	Johanna Koplik
05/07	Koosolek BaitPartner	1:30:00 15.00-16.30	Johanna Koplik
05/07	Disain BaitPartner	2:30:00 21.00-23.30	Johanna Koplik
05/08	Kirjandus BaitPartner	6:00:00 20.00-02.00	Johanna Koplik
05/09	Kirjandus BaitPartner	3:45:00 12.15-16.00	Johanna Koplik
05/09	Tiimi koosolek BaitPartner	1:00:00 15.00-16.00	Johanna Koplik
05/10	Kontainerite&kassetide loogika parandused BaitPartner	5:35:00 21.30-03.05	Johanna Koplik
05/11	Kontainerite&kassetide loogika parandused BaitPartner	3:50:00 14.00-17.50	Johanna Koplik
05/11	Tiimi kõne BaitPartner	1:05:00 22.00-23.05	Johanna Koplik
05/12	Kontainerite&kassetide loogika parandused BaitPartner	4:25:00 10.20-14.45	Johanna Koplik
05/12	Kirjandus BaitPartner	1:00:00 19.30-20.30	Johanna Koplik
05/13	Kirjandus BaitPartner	4:20:00 13.20-17.40	Johanna Koplik
05/16	Kirjandus BaitPartner	6:30:00 17.00-23.30	Johanna Koplik
05/17	Kirjandus BaitPartner	2:30:00 10.00-12.30	Johanna Koplik
05/17	Ettevõttega koosolek BaitPartner	0:30:00 16.00-16.30	Johanna Koplik
05/17	Clean code & kirjandus BaitPartner	6:00:00 17.00-23.00	Johanna Koplik

Joonis 24. Väljavõte ajakulust 5.

Linda Hmelevskaja

Detailed report			
Date	Description	Duration	User
01/29	Ettevõttega koosolek, projekti arutelu (no project)	1:30:00 14:30-16:00	Linda Hmelevskaja
02/01	Projekti arutelu (no project)	1:05:00 14:00-15:05	Linda Hmelevskaja
02/12	Ettevõttega koosolek (no project)	1:16:00 14:30-15:46	Linda Hmelevskaja
02/17	Tiimi koosolek (no project)	0:40:00 10:45-11:25	Linda Hmelevskaja
02/18	Lõputöö ülesandepüstitus (no project)	2:00:00 16:50-18:50	Linda Hmelevskaja
02/25	Lõputöö ülesandepüstitus (no project)	1:20:00 15:50-17:10	Linda Hmelevskaja
03/01	Projektiks ettevalmistumine koos tiimikaaslasega (no project)	4:25:00 12:00-16:25	Linda Hmelevskaja
03/01	Projekti seadistamine (no project)	6:50:00 17:07-23:57	Linda Hmelevskaja
03/02	Projekti seadistamine (no project)	9:17:00 09:08-18:25	Linda Hmelevskaja
03/03	Projekti seadistamine (no project)	9:54:00 09:15-19:09	Linda Hmelevskaja
03/04	Koodiga tutvumine (no project)	4:02:00 10:58-15:00	Linda Hmelevskaja
03/07	Kood (view model-ite ja view-de lisamine) (no project)	2:57:00 19:26-22:23	Linda Hmelevskaja
03/09	Meedia kuvamine (no project)	6:00:00 18:00-00:00	Linda Hmelevskaja
03/10	Meedia kuvamine, disain (no project)	1:20:00 15:15-16:35	Linda Hmelevskaja
03/11	Kood (no project)	3:57:00 15:46-19:43	Linda Hmelevskaja

Joonis 25. Väljavõte ajakulust 6.

03/13	Tiimi koosolek (no project)	1:45:00 12:00-13:45	Linda Hmelevskaja
03/19	Koosolek ettevõttega (no project)	0:56:00 12:00-12:56	Linda Hmelevskaja
03/24	Esimese blokki kood (no project)	4:33:00 13:00-17:33	Linda Hmelevskaja
03/25	Esimese blokki kood (no project)	3:33:00 10:44-14:17	Linda Hmelevskaja
03/31	Koosolek ettevõttega (no project)	0:55:00 14:00-14:55	Linda Hmelevskaja
04/01	Kirjandus (no project)	4:05:00 15:35-19:40	Linda Hmelevskaja
04/05	Tiimi koosolek (no project)	1:00:00 10:00-11:00	Linda Hmelevskaja
04/06	MeediaRepository kood (no project)	7:36:00 14:19-21:55	Linda Hmelevskaja
04/07	MeediaRepository kood (no project)	4:03:00 16:52-20:55	Linda Hmelevskaja
04/09	Kirjandus (no project)	4:40:00 15:54-20:34	Linda Hmelevskaja
04/10	Kirjandus (no project)	2:54:00 14:00-16:54	Linda Hmelevskaja
04/14	Kood (no project)	4:49:00 10:56-15:45	Linda Hmelevskaja
04/14	Dokumentatsioon, kirjandus (no project)	6:25:00 16:30-22:55	Linda Hmelevskaja
04/15	Test projekt (no project)	4:12:00 09:12-13:24	Linda Hmelevskaja
04/15	Kood, dokumentatsioon (no project)	5:11:00 12:44-17:55	Linda Hmelevskaja
04/15	Koosolek ettevõttega (no project)	1:10:00 14:25-15:35	Linda Hmelevskaja
04/16	Tiimi koosolek (no project)	1:00:00 22:00-23:00	Linda Hmelevskaja
04/18	Kood (konfliktide lahendamine, struktuuri muutmine, test projekt) (no project)	7:22:00 10:23-17:45	Linda Hmelevskaja

Joonis 26. Väljavõte ajakulust 7.

04/19	Tiimi koosolek (no project)	1:25:00 17:00-18:25	Linda Hmelevskaja
04/21	Dokumentatsioon (no project)	4:28:00 18:30-22:58	Linda Hmelevskaja
04/22	Dokumentatsioon (no project)	4:45:00 19:00-23:45	Linda Hmelevskaja
04/24	Dokumentatsioon (no project)	5:02:00 10:00-15:02	Linda Hmelevskaja
04/24	Dokumentatsioon (no project)	5:35:00 16:00-21:35	Linda Hmelevskaja
04/25	Dokumentatsioon (no project)	9:03:00 12:03-21:06	Linda Hmelevskaja
04/26	Tiimi koosolek (no project)	1:00:00 17:00-18:00	Linda Hmelevskaja
04/27	Dokumentatsioon (no project)	4:39:00 18:00-22:39	Linda Hmelevskaja
04/28	Dokumentatsioon (no project)	4:25:00 18:30-22:55	Linda Hmelevskaja
04/29	Dokumentatsioon (no project)	4:42:00 18:41-23:23	Linda Hmelevskaja
05/01	Dokumentatsioon (no project)	3:21:00 15:51-19:12	Linda Hmelevskaja
05/03	Meedia (no project)	3:33:00 20:12-23:45	Linda Hmelevskaja
05/04	Meedia (no project)	5:08:00 18:35-23:43	Linda Hmelevskaja
05/05	Testimine (no project)	5:27:00 18:30-23:57	Linda Hmelevskaja
05/06	Testimine (no project)	5:00:00 18:45-23:45	Linda Hmelevskaja
05/07	Juhendajatega kõne (no project)	1:15:00 15:00-16:15	Linda Hmelevskaja
05/07	Kood, dokumentatsioon (no project)	5:50:00 17:42-23:32	Linda Hmelevskaja
05/08	Kood (konteinerid) (no project)	2:00:00 10:06-12:06	Linda Hmelevskaja

Joonis 27. Väljavõte ajakulust 8.

05/08	Kood (konteinerid) (no project)	3:30:00 13:30-17:00	Linda Hmelevskaja
05/08	Kood (konteinerid) (no project)	4:38:00 18:07-22:45	Linda Hmelevskaja
05/09	Tiimi koosolek (no project)	1:00:00 15:00-16:00	Linda Hmelevskaja
05/09	Kood, dokumentatsioon (no project)	7:25:00 16:08-23:33	Linda Hmelevskaja
05/10	Dokumentatsioon (no project)	5:27:00 18:28-23:55	Linda Hmelevskaja
05/11	Save nupp, disain (no project)	3:34:00 18:26-22:00	Linda Hmelevskaja
05/11	Konteinerite teema arutelu (no project)	1:05:00 22:00-23:05	Linda Hmelevskaja
05/12	Kiiruse analüüs, dokumentatsioon (no project)	5:52:00 18:00-23:52	Linda Hmelevskaja
05/13	Dokumentatsioon (no project)	2:00:00 18:40-20:40	Linda Hmelevskaja
05/14	Testimine (no project)	0:48:00 18:46-19:34	Linda Hmelevskaja
05/15	Testimine, disain, kasutatud kassettide ikoon, puhas kood (no project)	3:47:00 10:00-13:47	Linda Hmelevskaja
05/15	Testimine, disain, kasutatud kassettide ikoon, puhas kood (no project)	7:35:00 15:12-22:47	Linda Hmelevskaja
05/16	Dokumentatsioon (no project)	7:42:00 14:48-22:30	Linda Hmelevskaja
05/17	Koosolek ettevõttega (no project)	0:30:00 16:00-16:30	Linda Hmelevskaja
05/17	Dokumentatsioon (no project)	4:32:00 19:00-23:32	Linda Hmelevskaja

Joonis 28. Väljavõte ajakulust 9.

4.4 Hinnang projekti teostamise protsessi kohta

Study Info projekt alustati 2020 sügissemestril kursuse ITB1706– Infosüsteemide arendamise meeskonnaprojekt raames ning kaitsti edukalt 2021 jaanuaris. Lõputööks andis Bait Partner OÜ võimaluse tõsta *Study Info* aken üle *Universal Windows* platvormile. Veebruaris alustati lõputöö projektiga ning teostati mai keskpaigaks. Projekti ülikooli poolseteks juhendajateks olid Kristina Murtazin ja Karl-Erik Karu. Ettevõtte poolt olid juhendajateks Inga Moreva, Brigita Urmet ja Jarmo Poolak. Suhtlus koolipoolsete juhendajatega toimus meili või MS Teamsi kaudu vastavalt vajadusele. Ettevõtte poolsete juhendajatega kohtuti MS Teamsi kõnedes ning suheldi Teamsi vahendusel. Oma tulemusi näidati videokõnes, kus saadi ka tagasisidet ja edasisi juhiseid.

Algselt töötati ja saadi kokku koolis, seda just projekti esimeses faasis. Hiljem tuli aga töötada kodunt ja ka omavahelisteks koosolekuteks kasutada MS Teamsi kõnesid. Ülesanded jaotati projekti algul vastavalt aknas kuvatavatele blokkidele. Töökäigus selgusid ülesannete rasked kohad ning sai vastavalt adapteeruda.

Kõige raskem projekti teostamise puhul oli, et me kumbki ei olnud ennem kokku puutunud UWP platvormiga ega projektide teisaldamisega uuele platvormile. Pidime algselt tutvuma nii uue platvormiga, kui ka teisaldusprotsessiga. Meie eeliseks oli, et teadsime varasema meeskonnaprojekti tõttu WPF-i kohta ja olime akna ise välja arendanud, kuid UWP jaoks oli vaja teha palju muudatusi. Projekti alustades võttis veel palju aega projekti *library*-te teisaldamine .NET Standardile, mis kokkuvõttes ei olnud vajalik ja saime koodi põhja ettevõttelt. Lisaks oli akna üks keerulisemaid osi konteinerite ja kassetidega seotud loogika, mille pidime uuesti ümber mõtestama ja välja töötama, sest ettevõtte uued kasutatavad projektid, ei olnud samasugused varasemalt WPF projektis kasutavale. Näiteks pidime andmebaasi päringute jaoks nüüd kasutama SQL lauseid. Samuti jäi kirjutatud koodi testimine projekti lõpufaasi ning selle tõttu ei jäänud ühiktestimiseks palju aega. Kogu aken sai aga testitud manuaalselt.

Lõputöö käigus saime ettevõtte tarkvarast palju süvendatumad teadmised, kui need olid meil peale meeskonnaprojekti ainet. See võimaldas teha parandusi akna üldises disainis ja funktsionaalsuses.

Projekti teostamine kulges ajaliselt väga lühikese perioodi jooksul. Töö oleks võinud jaotada pikema perioodi peale. Juhendajatega suhtlusest saadi kõik vajalikud suunised ja meeskonnaprojekt valmis suuremate probleemideta.

4.5 Meeskondlik hinnang panuse kohta

Meeskonnaliikmed panustasid projekti samaväärselt ning hindavad tööprotsessis osalemist hindegaga „0“ ehk „panustas samaväärselt“.

5 Kokkuvõte

Lõputöö eesmärkideks olid akna teisaldamine WPF töölauarakenduselt UWP-ga sobivaks, võrrelda WPF ja UWP rakendusi, nende sarnasusi ja erinevusi ning koostada juhend teisaldamiseks.

Projekti käigus tõsteti *Study Info* aken UWP platvormile ja sellel on valmisolek koos ettevõtte meditsiinirakendusega töötada kõigis Windows 10 operatsioonisüsteemidega seadmetes.

Projekti arendamiseks kasutati olemasolevat koodi, mille peale ehitati akna projekt. Lõputöö valmis kasutades C#, SQL ja XAML keeli. Tudengite töö toimus veebruarist mai keskpaigani.

Analüüsi käigus selgusid UWP eelised ja puudused võrreldes WPF projektidega. Anti edasi juhtnöörid teisaldamiseks UWP platvormile ning leiti meeskonnaprojektide käitusajaga võrreldes, et uus platvorm on tõepoolest kiirem ja paljulubav.

Study Info aken saadi sobivaks *Universal Windows*-i platvormiga, uuringu andmed kuvatud ja loogika testitud. Akna kujul sai meditsiinirakendus lisandi, mis näitab kogu uuringu informatsiooni ühest kohast ja oli nõutud kasutajate poolt.

Kasutatud kirjandus

- [1] „Bait Partner,“ [Võrgumaterjal]. Available: <https://www.baitpartner.eu/avaleht/>. [Kasutatud 28 04 2021].
- [2] J. Kouraklis, MVVM in Delphi: Architecting and Building Model View ViewModel Applications, Apress, 2019.
- [3] J. Albahari, C# 9.0 in a Nutshell, O'Reilly Media, 2021.
- [4] R. Hampton, „Here’s Why You Should Separate UI from Business Logic in Your Application,“ ICS, 2020. [Võrgumaterjal]. Available: <https://www.ics.com/blog/heres-why-you-should-separate-ui-business-logic-your-application>. [Kasutatud 01 05 2021].
- [5] Microsoft, „Walkthrough: Create and run unit tests for UWP apps,“ 2016. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/visualstudio/test/walkthrough-creating-and-running-unit-tests-for-windows-store-apps?view=vs-2019>. [Kasutatud 07 05 2021].
- [6] Microsoft, „Use code coverage for unit testing,“ 2021. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-code-coverage?tabs=windows>. [Kasutatud 07 05 2021].
- [7] C. Bernasconi, „Introduction to the Universal Windows Platform,“ ClaudioBernasconi, 2019. [Võrgumaterjal]. Available: <https://www.claudiobernasconi.ch/2019/02/28/introduction-to-the-universal-windows-platform-uwp/>. [Kasutatud 25 04 2021].
- [8] A. Chatterjee, Building Apps for the Universal Windows Platform: Explore Windows 10 Native, IoT, HoloLens, and Xamarin, Apress, 2017.
- [9] Microsoft, „Choose your Windows app platform,“ 2021. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/windows/apps/desktop/choose-your-platform>. [Kasutatud 28 04 2021].
- [10] K. Chowdhury, Mastering Visual Studio 2017, Packt Publishing, 2017.
- [11] Technopedia, „Native code,“ [Võrgumaterjal]. Available: <https://www.techopedia.com/definition/3846/native-code>. [Kasutatud 29 04 2021].
- [12] K. Gregory, „Managed, Unmanaged, Native: What Kind of Code Is This?,“ [Võrgumaterjal]. Available: <https://www.developer.com/microsoft/c-sharp/managed-unmanaged-native-what-kind-of-code-is-this/>. [Kasutatud 28 04 2021].
- [13] „Overview of XAML,“ Microsoft, 2020. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/visualstudio/xaml-tools/xaml-overview?view=vs-2019>. [Kasutatud 27 04 2021].
- [14] S. Basu, „Why XAML? Why Now?,“ ProgressTelerik, 2018. [Võrgumaterjal]. Available: <https://www.telerik.com/blogs/why-xaml-why-now>. [Kasutatud 25 04 2021].

- [15] A. Nathan, XAML Unleashed, Sams, 2014.
- [16] Microsoft, „DockPanel,“ 2021. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/windows/communitytoolkit/controls/dockpanel>. [Kasutatud 28 04 2021].
- [17] Microsoft, „Windows Community Toolkit Documentation,“ 2021. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/windows/communitytoolkit/>. [Kasutatud 25 04 2021].
- [18] Microsoft, „Deploy UWP Apps,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/visualstudio/debugger/deploy-windows-store-apps-from-visual-studio?view=vs-2019>. [Kasutatud 28 04 2021].
- [19] Microsoft, „Deploy a WPF Application,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/app-development/deploying-a-wpf-application-wpf?view=netframeworkdesktop-4.8>. [Kasutatud 28 04 2021].
- [20] R. Fakaliev, „Building Modern & Performant Desktop Apps—Is WinUI 3.0 the Way to Go?,“ ProgressTelerik, 2020. [Võrgumaterjal]. Available: <https://www.telerik.com/blogs/building-modern-performant-desktop-apps-winui-30-the-way-to-go>. [Kasutatud 25 04 2021].
- [21] „Choose a UWP version,“ Microsoft, 2019. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/windows/uwp/updates-and-versions/choose-a-uwp-version>. [Kasutatud 25 04 2021].
- [22] A. Jana, „Selecting right version while creating your Universal Windows Platform (UWP) App,“ Daily .NET Tips, 2016. [Võrgumaterjal]. Available: <https://dailydotnettips.com/selecting-right-version-while-creating-your-universal-windows-platform-uwp-app/>. [Kasutatud 25 04 2021].
- [23] Microsoft, „Data access,“ 2017. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/windows/uwp/data-access/>. [Kasutatud 28 04 2021].
- [24] Microsoft, „.NET Native – What it means for Universal Windows Platform (UWP) developers,“ 2015. [Võrgumaterjal]. Available: <https://blogs.windows.com/windowsdeveloper/2015/08/20/net-native-what-it-means-for-universal-windows-platform-uwp-developers/>. [Kasutatud 28 04 2021].
- [25] Microsoft, „Move from a desktop application to UWP,“ 2017. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/windows/uwp/porting/desktop-to-uwp-migrate>. [Kasutatud 28 04 2021].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Linda Hmelevskaja

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Meditiinirakenduse akna üleviimine *Universal Windows* platvormile“, mille juhendajad on Kristina Murtazin ja Karl-Erik Karu
 - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

17.05.2021

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Enda panuse kirjeldus ja eneseanalüüs, Johanna Maria Koplík

Projekti algul leppisime kokku ülesande jaotuses, kus minule jäid teha osad seotud üld- ja patoloogilise uuringu infoga. Saime osad nii jaotada, sest akna *front-end* arendusel jagunes vaade blokkideks ja ka *back-end*-is ei sõltunud need osad üksteisest. Täpsemalt tegelesin väljadega *Start time*, *End time*, *Study id*, *Procedure name*, *Procedure code*, *Performer*, *Assistent*, *Total quantity of Containers*, *Total quantity of Cassettes*, *Cassette Size*, *Macroscopic description* ja *Preliminary description*. Akna keskmise osa, konteinerite ja kassettidega seotud loogika planeerisime arendada koos. Töö käigus kujunes minu tööks suuremas osas veel disainime, kuna konteinerite ja kassettide esitamine oli tihedalt sellega seotud, tegelesin ka palju konteinerite ja kassettide kuvamisega. Konteinerite ja kassettide osa all olevad väljad, millega tegelesin on järgmised: *Specimen size*, *Cassette no*, *Cassette Size*, *Used cassettes*, *Ordered cassettes*, *Added cassettes* ja *Cassette Count*. Lisaks projekti alguses tõstsingi WPF projekti eeskujul projektid ja klassid üle .Net Standard *library*-te peale. Ettevõtte muutuste tõttu süsteemi arhitektuuris, mis nad juba UWP platvormiga olid sobivaks teinud, soovitasid nad ka meil kasutada nende süsteemi arhitektuuri ja see töö kasutusse ei läinud.

Kaheliikmelises tiimis töötamisel pidime projekti panustama mõlemad algusest peale ning ei jäänud ruumi liigselt teiste peale lootmiseks. Lisaks koolis tehtud meeskonnatöödele sain nüüd näha meeskonnatööd päris ettevõttes, mille tööprotsess erines mõnevõrra varasemast ja sai ka kasutada koolis õpituid meetodeid. Projekti käigus sain paremaks ettevõttega suhtlemisel ja enda arendustööde esitamisel. Bait Partner OÜ-s meeskonnaprojekti aine raames alustatud ja lõputööna lõpetatud *Study Info* projekti arendamine on andnud mulle suure kogemuse päris elu arendusprotsessist ja nende eri etappidest. Tean nüüd, kui oluline on ka kliendi vajaduste mõistmine arenduses.

Projekti käigus oli suurimaks väljakutseks UWP raamistik, millega ei olnud varem me kumbki kokku puutunud. Raskeimaks osaks arendamisel kujunes konteinerite ja kassettide kuvamine aknale, mida sain mitu korda ümber teha. Pidin palju juurde õppima

ja mõtlesime töökäigus ka mõned asjad ümber. Lõputöö arendamisel aitas kaasa, et olime ise *Study Info* akna WPF-is arendanud ja saime palju tuge oma varasemast lahendusest. Oleksime võinud keskenduda varem testimisele, sest see jäi üpris lõpufaasi. See oli aga tingitud UWP raamistikust, mis ei olnud meile algselt tuttav ja me ei teadnud ning ei leidnud pikalt sobivat varianti, mida saab testimiseks kasutada.

Lisa 3 – Enda panuse kirjeldus ja eneseanalüüs, Linda Hmelevskaja

Alguses ülesannete jagamisel jäi minule teha osad seotud patsiendi, üld- ja patoloogilise uuringu ning meedia andmetega. Täpsemad väljad, millega ma tegelesin olid kõik patsiendi andmed, ehk *Patient first name*, *patient last name* ja sellest tulenev *Patient full name*, *Patient SSN* ja *Patient age*. Uuringu andmeteks olid *Study category level*, *Contamination warning*, *Warning for needles*. Meedia puhul samuti kõik andmed, ehk *List of recorded images*, *List of recorded videos*, *List of recorded audio*, *Files sent to PACS (No. and time)*. Samuti aitasin konteinerite ja kassettide kuvamisega. Lisaks sellele tegelesin disainiga.

Üks minu ülesannetest oli ka ühiktestimine. Otsides parima variandi UWP rakenduse testimise jaoks, ettevõttepoolne juhendaja soovitas muuta struktuuri ja tõsta vaadete mudelid *Facade* kausta. Ühiktestimise jaoks olid proovitud erinevad testimise projektid, kuid lõpuks otsustati kasutada spetsiaalset UWP ühiktestide projekti. Selle kohta on rohkem informatsiooni, mis aitas kiirendada ühiktestide kirjutamise protsessi. Probleemiks aga osutus see, et testimisega alustasime hiljem ja sellele ei jäänud väga palju aega.

Raskeimaks osaks oli kindlasti projektiga alustamine. Kuna UWP-ga ei olnud enim kokku puutunud, siis võttis aega antud platvormi kohta uurimine. Projekti arendamist lihtsustas kindlasti see, et kooli ja töö projektide käigus saadud C# kogemus oli piisavalt suur. Probleeme tekitas ka projekti seadistamine vale Windows versiooni pärast isiklikus arvutis. See lükkas projekti arendamist edasi, kuna võttis päris palju aega.

Lisaks sellele oli ka konteinerite ja kassettide kuvamine üheks väljakutseks. See ei olnud mu põhiülesanne, aga kuna sügise meeskonnaprojekti ajal ma ei tegelenud selle osaga üldse, siis pidi palju uurima ja õppima juurde.

Nii sügise meeskonnaprojekti kui ka lõputöö projekti raames kõige arendavam oli reaalne töö kliendiga. Selle käigus õppisin mitte ainult seda, kuidas UWP või WPF

rakendusi arendada. Tähtsaks kogemuseks oli ka pidev suhtlemine ja arutelu kliendiga. Kliendil võivad olla kindlad nõuded ja soovid, kuid võivad tekkida olukorrad, kus ta ise ei ole kindel, mida ta tahaks või kuidas see peaks välja nägema. Sel juhul on oluline ise analüüsida ja näidata, mis võivad olla alternatiivid.

Koostöö ettevõttega oli tõhus ja sujuv. Alati kui tekkis mingi takistus seoses koodi kirjutamisega või disainiga olid juhendajad valmis aitama. Pidev koostöö oli tähtis selle jaoks, et tehtud aken oleks sobiv ettevõtte tarkvaraga. Niimoodi saime luua väärtust ettevõtte jaoks piisavalt kiiresti.

Lisa 4 – *Study Info* akna vaade



Start:

End:

 44 Y
3 M
16 D





Study id:

Procedure name:

Procedure code:

Performer:

Assistant:

Patient name:

Patient SSN:

Comment:

Comments typically include relevant information that is not a diagnosis.

Sample info	Macroscopic description	Preliminary description
Total quantity:	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <input type="text" value="6"/></div> <div style="text-align: center;"> <input type="text" value="13"/></div> <div style="text-align: center;"> <input type="text" value="1"/></div> <div style="text-align: center;"><input type="text" value="12"/></div> </div>	
> 1	3 cassettes	used 3 / 0 ordered
> 2	2 cassettes	used 1 / 0 ordered
> 3	2 cassettes	used 2 / 0 ordered
> 4	3 cassettes	used 2 / 0 ordered
> 5	3 cassettes	used 3 / 0 ordered
> 6	0 cassettes	used 0 / 0 ordered

Recorded files:

 34

 4

 2

Sent to archive:



2 / 10-Oct-16 15:47:12



Joonis 29. *Study Info* akna vaade 1.



Start: 16.11.2015 15:00
 End: 16.11.2015 17:00

44 Y
 3 M
 16 D



Study id: XYZ-1123456
 Procedure name: Gastroscopy's sample
 Procedure code: Endoscopy
 Performer: Gregory House
 Assistent: John Smith

Patient name: Diana Hopkins
 Patient SSN: 725105
 Comment: Comments typically include relevant information that is not a diagnosis.

Sample info

Macroscopic description

Preliminary description

Macroscopic is the opposite of microscopic, which describes anything you need a microscope to see. A scientist might use macroscopic to talk about the regular view of an object, when you're looking at it unaided by any magnification device.

	Cassettes	Used / Ordered	Added
>	1 3 cassettes	used 3 / 0 ordered	3 added
∨	2 2 cassettes	used 1 / 0 ordered	2 added

Specimen size	No	Size	Used										
<table style="width: 100%; border-collapse: collapse;"> <tr> <td>L</td><td>W</td><td>H</td><td>A</td><td>D</td> </tr> <tr> <td>4.30000</td><td>4.30000</td><td>12.40</td><td>15.20</td><td>5.50</td> </tr> </table>	L	W	H	A	D	4.30000	4.30000	12.40	15.20	5.50	1	Normal	✖
L	W	H	A	D									
4.30000	4.30000	12.40	15.20	5.50									
	2	Normal											



Recorded files:

 34

 4

 2

Sent to archive:

 2 / 10-Oct-16 15:47:12



Joonis 30. Study Info akna vaade 2.