

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Lisette Noor 175254IDDR

# **Rakendus kinokülastuse planeerimiseks**

Diplomitöö

Juhendaja: Meelis Antoi

Magistrikraad

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: Lisette Noor

29.04.2021

## **Annotatsioon**

Käesoleva diplomitöö eesmärgiks on luua rakendus, mis kuvaks informatsiooni Eestis olevate kinokavade kohta ning hõlbustaks kinokülastuste planeerimist.

Töös tehakse ülevaade eksisteerivatest lahendustest ning määratakse nõuded, mida loodav rakendus peab täitma.

Projekti arenduse käigus kirjutatakse programm, mis eraldab vajalikke andmeid veebilehekülgedelt, seatakse üles andmebaas ning luuakse rakendus, mis näitab ajakohast informatsiooni kinokavade kohta ja teavitab, kui lisatud on uus pealkiri.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 24 leheküljel, 7 peatükki, 16 joonist, 2 tabelit.

## **Abstract**

### **Application for Planning Cinema Visits**

The aim of this thesis is to create an application which displays information about all the current movie schedules in Estonian cinemas and to make planning cinema visits more convenient.

The thesis provides an overview of existing solutions and the requirements that the developed application has to implement.

During the development a program is created which gathers the required information from websites, a database is set up to store that information and an application is created which displays up-to-date information about movie schedules and notifies the user if a new entry was added to the schedule.

The thesis is in Estonian and contains 24 pages of text, 7 chapters, 16 figures, 2 tables.

## Lühendite ja mõistete sõnastik

AJAX	<i>Asynchronous JavaScript And XML</i> , veebiarenduse tehnoloogiate kogum, mis võimaldab asünkroonseid päringuid
Android	Operatsioonisüsteem, mida kasutavad telefonid, tahvelarvutid ja nutikellad
CSV	<i>Coma-separated values</i> , tekstifail, kus väärtused on komaga eraldatud
Excel	Tabelarvutussüsteem
HTML	<i>Hyper Text Markup Language</i> , veebilehede märgistuskeel
Integratsioonitestimine	Tarkvaramoodulite koostöö testimine, kus kontrollitakse, kas kokku pandud moodulid töötavad omavahel
JSON	<i>Javascript Object Notation</i> , andmete formaat
JVM	<i>Java virtual machine</i> , Java virtuaalmasin, mis tõlgendab Java koodi arvutile arusaadavateks juhenditeks
Moodulitestimine	Tarkvaramooduli testimine, kus kontrollitakse, kas moodulis olevad funktsioonid töötavad korrektselt
PHP	<i>Hypertext Preprocessor</i> , skriptimiskeel, mida peamiselt kasutatakse serveripoolse veebilehe loomiseks
Süsteemtestimine	Süsteemi kui terviku testimine „musta kasti“ meetodil ehk vaadeldakse kasutajale nähtavaid osi süvenemata koodile
URL	<i>Uniform Resource Locator</i> , internetiaadress
USB	<i>Universal Serial Bus</i> , seadmete ühendamise standard
Veebikaabits	Veebikoorimise ehk veebilehelt andmete eraldamise programm või moodul

## Sisukord

Autorideklaratsioon .....	2
Annotatsioon.....	3
Abstract Application for Planning Cinema Visits .....	4
Lühendite ja mõistete sõnastik .....	5
Sisukord .....	6
Jooniste loetelu .....	8
Tabelite loetelu .....	9
1 Sissejuhatus .....	10
2 Metoodika.....	11
3 Ülevaade probleemist .....	12
3.1 Eksisteerivad lahendused.....	12
3.1.1 Forum Cinemas EE.....	12
3.1.2 Apollo Cinema.....	12
4 Loodava rakenduse analüüs.....	13
4.1 Nõuded.....	13
4.2 Tarkvara valik.....	14
4.2.1 Veebikaabits .....	14
4.2.2 Andmebaas .....	15
4.2.3 Programmeerimiskeeled .....	16
4.3 Arhitektuur.....	18
4.4 Analüüsi kokkuvõte .....	19
5 Töö lahendus .....	21
6 Testimine .....	28
6.1 Moodultestimine.....	28
6.2 Integratsioonitestimine .....	28
6.3 Süsteemtestimine.....	29
7 Kokkuvõte .....	30
Kasutatud kirjandus .....	31

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	33
Lisa 2 – Rakenduse vaated .....	34

## Jooniste loetelu

Joonis 1. Rakenduse arhitektuur.....	18
Joonis 2. Andmebaasi tabel.....	19
Joonis 3. Veebikaabitsa skeem.....	19
Joonis 4. Andmete eraldamisel kasutatud URL-id.....	21
Joonis 5. Kinokava HTML.....	21
Joonis 6. Andmete eraldamine XPath-iga .....	22
Joonis 7. MySQL-ga ühenduse loomine .....	23
Joonis 8. Saadud andmed MySQL andmebaasis.....	23
Joonis 9. Andmebaasiga ühendamine ja tagastamine JSON formaadis.....	24
Joonis 10. Androidi pääsuluba.....	24
Joonis 11. Rea disain.....	24
Joonis 12. Andmete lugemine serverist.....	25
Joonis 13. <i>TextView</i> atribuudid.....	26
Joonis 14. Serverist saadud andmete määramine <i>TextView-dele</i> .....	26
Joonis 15. Meetod veebi brauseri avamiseks.....	27
Joonis 16. Aja kontrollimise blokk skeem .....	27



## **Tabelite loetelu**

Tabel 1. Andmebaaside võrdlus.....	16
Tabel 2. Programmeerimiskeelte võrdlus.....	17

## 1 Sissejuhatus

Kinos käimine ja filmide vaatamine on Eestis kujunenud populaarseks ajaviiteks – kinoasutused korraldavad küllastajatele eriseansse ning filmidega seotud üritusi nagu näiteks PÖFF ehk Pimedate Ööde filmifestival ja HÖFF ehk Haapsalu õudus- ja fantaasiafilmide festival.

Jooksvate ja peatselt linastuvate filmide kohta saab teavet mitmel erineval viisil. Näiteks kinoasutuste kodulehtede küllastamine ning veebilehelt kavade vaatamine. Suurfilmide puhul pannakse üles välireklaamid ning näidatakse reklaamklippe televisiooni kommertskanalites ja saadetakse teavitusi uudiskirjade kaudu. Peatselt linastuvate filmide kohta saab teada ka reklaamklippide kaudu, mida näidatakse kinosaalides enne filmi algust. Selleks, et õigel ajal kinopileteid osta saaks tuleb filmi kuupäeva meeles pidada, korduvalt kontrollida kinokavasid või omale kalendrisse märkida.

Käesolev lõputöö analüüsib olemasolevaid lahendusi Eesti kontekstis kuidas kinokavade kohta teavet saadakse. Antud probleemi lahenduseks on pakutud rakendus, mis näitab ajakohast informatsiooni kinokavade kohta ja teavitab, kui uus film on lisatud jooksvate filmide kavasse. Samuti saab kasutaja valida, milline on eelistatud kinoasutus ning žanrid.

Diplomitöö autor on samuti kinoküllastaja ning käib tihti seltskonna või perega kinos filme vaatamas. Korduvalt on olnud olukordi, kus lõputöö autor on näinud reklaamklippi tulevast filmist, kuid on ajapikku unustanud, mis kuupäevast saab seanssidele pileteid osta. Mõnel juhul on kinokavas olnud vähem reklaamitud film, mille vastu on huvi tundnud, aga pole selle kohta teavet saanud ja on vaatamisvõimaluse maha maganud. Seega leiab autor, et antud projekt lahendab püstitatud probleemi.

## **2 Metoodika**

Diplomitöö käigus selgitatakse lahti antud probleem, eksisteerivad lahendused ning nende omadused ning pakutakse vastav IT lahendus, arvestades piiratud aega projekti tegemiseks. Lahendust on tulevikus võimalik edasi arendada.

Loodava lahenduse analüüsi käigus tuuakse välja rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded. Samuti analüüsitakse valitud tehnoloogiaid, mis aitavad antud lahenduseni jõuda.

Lahenduse valmimist on kirjeldatud osadena, mis käsitlevad kinokavade info kogumist, andmebaasi ja serveri lahendust ning rakenduse enda valmimist. Lõpuosas on kirjeldatud lahenduse testimist ning edaspidiseid arendusvõimalusi.

### **3 Ülevaade probleemist**

Kinokavade jälgimiseks peab külastama eraldi kinode kodulehti veebis või käima läbi mobiilirakendused samuti saab uudiskirjade kaudu teateid ainult kinoasutustes toimuvate ürituste ja eriseansside kohta ning antud juhtudel peab ise meeles pidama või meeldetuletuse oma kalendrisse lisama, mis kuupäeval oodatud film kinokavasse lisatakse. Praegusel koroonaaajal lükatakse esilinastusi tihti edasi ja kalendris tuleb infot vastavalt ka uuendada. Oleks vaja sobivat IT lahendust, kus kasutaja saab mugavalt ühest kohast vaadata, mis filmid on antud hetkel kinokavas ja selle järgi planeerida kinokülastusi. Antud probleemiks on olemas lahendused, kuid nende ühiseks omaduseks on see, et lahendused käsitlevad ainult ühte konkreetset kinoasutust.

#### **3.1 Eksisteerivad lahendused**

##### **3.1.1 Forum Cinemas EE**

Forum Cinemas EE on Forum Cinemas OÜ enda mobiilirakendus, mis osaliselt lahendab püstitatud probleemi. Nimelt on rakenduses võimalik vaadata kinokavasid, luua endale kasutajakonto ning osta ka pileteid seanssidele, kuid antud rakendus hõlmab ainult Forum Cinemas kinoasutust, samuti puudub rakendusel teavituste valik.[1]

##### **3.1.2 Apollo Cinema**

Sarnaselt Forum Cinemas EE rakendusele on Apollo Cinema mobiilirakendus, mis hõlmab ainult Apollo kinosid. Samuti on rakenduses võimalik luua endale kasutajakonto, osta kinopileteid ja saada informatsiooni Apollo kinokavade kohta üle Eesti, kuid rakendusel pole võimalik teavitusi seadistada.[2]

## 4 Loodava rakenduse analüüs

### 4.1 Nõuded

Antud alapeatükis kirjeldatakse kasutajalugudena, milliseid funktsionaalseid ja mittefunktsionaalseid nõudeid peab antud projekt täitma. Kasutajate põhirollideks on kinohuviline ehk tavakasutaja ja administraator.

Nõuded, mida soovitakse edasi arendada peale praeguse prototüübi valmimist, käsitletakse peatüki lõpus.

Tavakasutaja funktsionaalsed nõuded:

- Tavakasutajana soovin näha ajakohast kinokava.
- Tavakasutajana soovin teada saada, kui uus film on lisatud kavasse.
- Tavakasutajana soovin valida, millise kino kava vaadata.
- Tavakasutajana soovin näha filme valitud žanri järgi.
- Tavakasutajana soovin, et rakenduse kaudu saan avada veebilehe, kust saan pileteid osta.
- Tavakasutajana soovin saada meeldetuletust kui järgmisel päeval on kinnominek.

Tavakasutaja mittefunktsionaalsed nõuded:

- Tavakasutajana soovin kasutada rakendust mobiiltelefoni peal.
- Tavakasutajana soovin aru saada rakenduse vaadete sisust.

Administraatori mittefunktsionaalsed nõuded:

- Administraatorina soovin, et koodibaas oleks rahvusvahelises keeles juhul kui rakenduse arendusse kaasatakse rohkem inimesi.

Arvestades lõputöö tegemise ajapiiranguga on mõned planeeritavad funktsionaalsused jäetud väljapoole töö skoopi. Nende funktsionaalsuste hulka kuulub näiteks kalendri kuvamine neile, kes kasutavad kalendri rakendust. Funktsioon seisneb selles, et uue filmi lisandumisel edastatakse teavitust ning kuvatakse ka informatsioon, mis on kalendris samal kuupäeval märgitud.

## **4.2 Tarkvara valik**

### **4.2.1 Veebikaabits**

Veebikaabitsate ehk veebileheküljelt andmete eraldamiseks mõeldud programmi kirjutamiseks on palju võimalusi. On võimalik kasutada tööriistu nagu ParseHub ja OctoParse või arendada ise vastav programm kasutades teeki nagu Selenium ja BeautifulSoup või Scrapy raamistikku. Antud alapeatükis tehakse ülevaade pakutud võimalustest.

ParseHub on tööriist, mille abil saab luua veebikaabitsat ilma koodi kirjutamata. Tööriista kasutajaliideses sisestatakse URL, kust soovitakse andmeid eraldada ning käsitsi valitakse vajalikud andmeväljad. Tööriista käivitamisel võetakse veebileheküljelt andmed ning esitatakse eraldatud andmed JSON või CSV formaadis või Excel tabelina. Samuti on võimalik saata saadud andmed andmebaasi.[6]

OctoParse on sarnaselt ParseHub-le tööriist, mille abil saab eraldada veebilehekülgedelt andmeid ilma, et peaks ühtegi rida koodi kirjutama. Kasutajasõbralikus liideses sisestatakse URL ning veebilehekülje sisu kuvatakse tööriista vaates. Seal valitakse teksti- ja pildiväljad, mida soovitakse kasutada. Seejärel tööriista käivitamisel konverteeritakse valitud andmed CSV formaati, Excel tabeliks või saadetakse andmebaasi.[7]

Tööriistade kasutamise suurimateks eelisteks on aja kokkuhoid töö lahendamise arendamise mõistes, kuid nende kasutusvõimalused on piiratud seoses sellega, et need on tasulised. Antud tööriistadel on küll võimalik valida tasuta variant, kuid need on mõeldud selleks, et klient saaks testida tööriista kasutatavust ning antud variandid ei paku automatiseerimist. Seega ei ole võimalik määrata ajagraafikut, mille järgi tööriistad

eraldavad ja tagastavad andmeid selleks, et andmebaasis oleks ajakohane informatsioon. Seepärast peab autor kasutama avatud lähtekoodiga teeke ja raamistikke.

Veebilehekülgedelt informatsiooni lugemiseks ja kopeerimiseks HTML-st on võimalik kasutada teeke nagu Selenium ja BeautifulSoup või Scrapy raamistikku. Neid kasutatakse peamiselt programmeerimiskeeles Python.

Scrapy on Pythoni programmeerimiskeeles kirjutatud raamistik, mille abil eraldatakse andmeid veebist [3].

Scrapy suurimaks eeliseks on kiirus ning sisseehitatud tugitarkvara, mis võimaldab veebilehekülgedel HTML-is navigeerida ning vastavaid andmeid eraldada.

Antud teegi kiirus on tingitud sellest, et Scrapy on ehitatud Twisted raamistiku peale, mis lubab asünkroonselt saata päringuid veebiserverisse [4].

Võrreldes Scrapyga on BeautifulSoup ja Selenium küll algajasõbralikumad, kuid nende puhul on vaja lisa teeke selleks, et veebileheküljelt alla laetud andmeid töödelda.

#### **4.2.2 Andmebaas**

Andmebaasi valimise käigus tuleb arvestada andmete töötlemise kiiruse ja eeldatava andmete hulgaga. Järgnevalt on vaatluse alla võetud laialt kasutatud ja avatud lähtekoodiga andmebaasid ning analüüsitud nende erinevusi.

MySQL on relatsiooniline andmebaas, mis lubab hoida suurt hulka andmeid ning töötleb andmeid kiiresti [8]. Samuti on võimalik andmebaas ühendada serveriga juhul kui andmebaasi hoitakse rakendusest eraldi ning andmete edastamine toimub üle võrgu [9].

SQLite on piiratud suurusega, kuid kiire relatsiooniline andmebaas. Andmebaasi kasutamine on kergesti õpitav ning seda kasutatakse juhul kui luuakse rakendus, mille andmebaasi hoitakse rakenduse enda sees ning nende vahel toimub andmevahetus. Lisaks piiratud mahule puudub SQLite serveriga ühendamine [9]. Androidis on vaikimisi SQLite implementeeritud, seega ei oleks vajalik andmebaasi üles seada [5].

PostgreSQL on skaleeritav relatsiooniline andmebaas, mis on samuti suure kasutajabaasiga nagu MySQL. Võrreldes MySQL andmebaasiga peab PostgreSQL rohkem kinni SQL reeglitest, kuid operatsioonide täitmine ei ole kõige efektiivsem [9].

Järgnevalt on välja toodud nimetatud andmebaaside võrdlus, kus võrdluspunktideks on nende eelised ja puudused (Tabel 1).

<b>Andmebaas</b>	<b>Eeldused</b>	<b>Puudused</b>
MySQL	Kergesti õpitav Kiire	Osaliselt avatud lähtekoodiga
SQLite	Kiire Kergesti õpitav	Piiratud andmebaasi suurus
PostgreSQL	Skaleeritav Avatud lähtekoodiga	Puudulik dokumentatsioon

Tabel 1. Andmebaaside võrdlus

#### 4.2.3 Programmeerimiskeeled

Tööks vajalike tehnoloogiate valik sõltub paralleelselt programmeerimiskeelte valikust kuna erinevaid tehnoloogiaid rakendatakse erinevates keeltes. Järgnevalt on välja toodud iga tehnoloogia jaoks programmeerimiskeelte variandid, mida saaks kasutada prototüübi arendamisel ning nende erinevused.

Python on neist kõige populaarsem keel veebikaabitsa loomiseks. See on kergesti õpitav, kõrgetasemeline interpreteeritav objekt-orienteeritud programmeerimiskeel, mida saab kasutada ka skriptimiseks [11]. Pythoni eeliseks on see, et kõik andmete eraldamisega seotud protsessid töötavad kiiresti. Samuti on võimalik kasutada teeki nagu Beautiful Soup ja Scrapy raamistikku, mis hõlbustab programmi võimet eraldada andmeid tõhusalt [13].



Javascript on objekt-orienteeritud programmeerimiskeel, mida kasutatakse peamiselt veebilehtede skriptimiseks [14]. Antud keelt eelistatakse juhul kui on tegemist veebilehega, mis on dünaamiliselt genereeritud ning eraldatavate andmete hulk on väike [13].

Ruby on interpreteeritav üldotstarbeline programmeerimiskeel, millel on kergesti õpitav süntaks ja suur kogus teeke, mis hõlbustab veebilehe töötlemist [15]. Võrreldes eelnimetatud keeltega on Ruby keeles kirjutatud veebikaabitsad aeglasemad ning antud keeles saab eraldada andmeid ainult serveri poolelt – kui veebilehe sisu kuvamiseks kasutatakse AJAXit, tuleb sel juhul lisa teeke ja programme kasutusele võtta [13].

Java on objekt-orienteeritud programmeerimiskeel, mida on võimalik kasutada mitmetel platvormidel. Samuti on see üks populaarseimatest keeltest maailmas, mida kasutatakse rakenduste arendamiseks [12]. Java on hästi dokumenteeritud ning kasutatakse vaikumisi Androidi rakenduste loomiseks [16].

Kotlin on avaliku lähtekoodiga staatiliselt tüübitud programmeerimiskeel, mis esialgu loodi JVMi ja Androidi jaoks kombineerides objekt-orienteeritud ning funktsionaalset programmeerimist. Võrreldes Java keelega on Kotlin uuem ning selle süntaks kokkuvõtlikum.[17]

Järgnevalt on välja toodud programmeerimiskeelte võrdlus kasutades võrdluspunktideks lõputöö autori kogemust ning keele õppimiskeerukust (Tabel 2).

<b>Keel</b>	<b>Kogemus</b>	<b>Õppimiskeerukus</b>
Python	Väga hea	Madal
Javascript	Vähene	Madal
Ruby	Puudub	Madal
Kotlin	Puudub	Keskmine
Java	Rahuldav	Keskmine

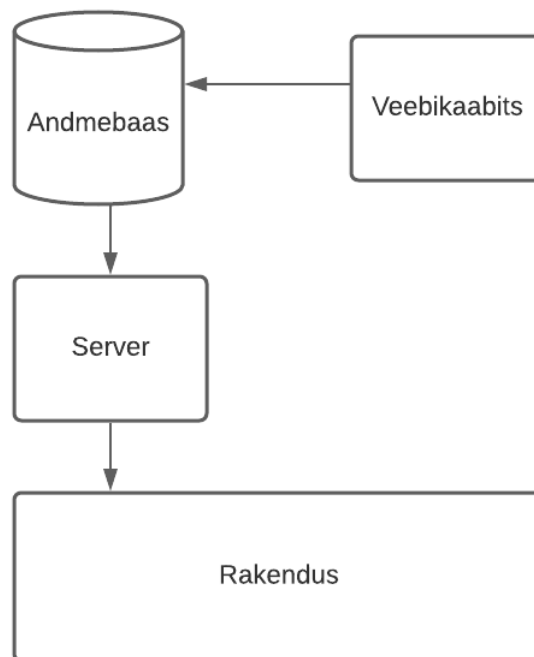
Tabel 2. Programmeerimiskeelte võrdlus

### 4.3 Arhitektuur

Rakenduse üldine arhitektuur koosneb järgnevatest komponentidest:

- Veebikaabits
- Server
- Andmebaas
- Rakendus

Joonisel 1 on välja toodud antud komponentide omavaheline suhe. Joonisel 2 on kasutatud andmebaas:

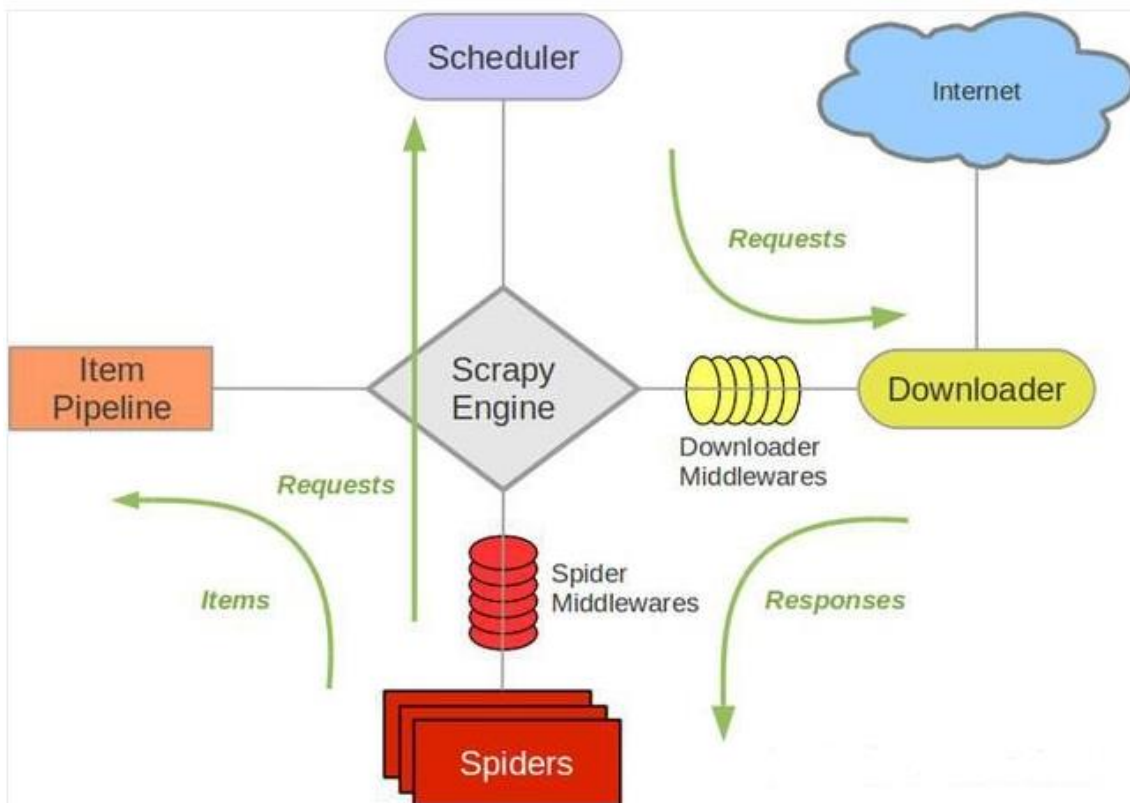


Joonis 1. Rakenduse arhitektuur

movies	
id	int
cinema	varchar
schedule_type	varchar
movie	varchar
original_title	varchar
date	date
genre	varchar
runtime	varchar
movie_url	varchar

Joonis 2 Andmebaasi tabel

Joonisel 3 on välja toodud veebikaabitsa skeem:



Joonis 3 Veebikaabitsa skeem

#### 4.4 Analüüsi kokkuvõte

Projekti planeerimise käigus jõuti järeldusele, et antud lahenduse loomiseks on vaja programmi, mis kogub kõikidelt kino veebilehekülgedelt kava info. Järgmisena on tarvis

andmebaasi ning andmete töötlemiseks seada üles server. Lõpuks tuleb luua rakendus, mis kuvab kasutajale vastavat informatsiooni.

Analüüsis sai käsitletud võimalike tehnoloogiate valikuid, mille abil saab luua lahenduse prototüübi, samuti toodi välja tehnoloogiate eelised ja nende puudused ning seejärel analüüsiti programmeerimiskeeli, mida kasutatakse nimetatud tehnoloogiates.

Veebikaabitsa loomiseks valiti programmeerimiskeeleks Python ja kasutati Scrapy raamistiku. Analüüsist järeldades on Pythonis kirjutatud Scrapy veebikaabits kõige tõhusam pidades silmas programmi kiirust ning töödeldavate andmete mahtu. JavaScript ja Ruby on küll võrdlemisi kergesti õpitavad, kuid lõputöö autoril on juba pikaajaline kogemus Pythoni programmeerimiskeelega ning Scrapy raamistiku kasutamisega.

Arvestades seda, et andmete maht tuleb üsna suur ning projekti edasiarendamisel läheb maht veelgi suuremaks ja rakendus tehakse Androidile, valiti andmebaasiks MySQL. Androidis on küll vaikumisi SQLite implementeeritud, kuid antud juhul SQLite ei ole sobiv variant, kuna tegemist on piiratud suurusega andmebaasiga. Antud töö jaoks on vaja suurt andmebaasi, mida hoitakse projektivälisena. Samuti hoitakse selle arvelt kokku ressursse.

Rakenduse loomiseks valiti Java, sest antud lõputöö eesmärgikson luua rakendus, mida saab kasutada mobiiliseadmes ning Androidis kasutatakse Javat vaikumisi. Kotlini keeles on ka võimalik kirjutada Androidi rakendusi, kuid see on võrdlemisi uus programmeerimiskeel ja enamus Android-i koodinäited on Java keeles saadaval. Samuti puudub töö autoril kogemus Kotlini kasutamisel.

## 5 Töö lahendus

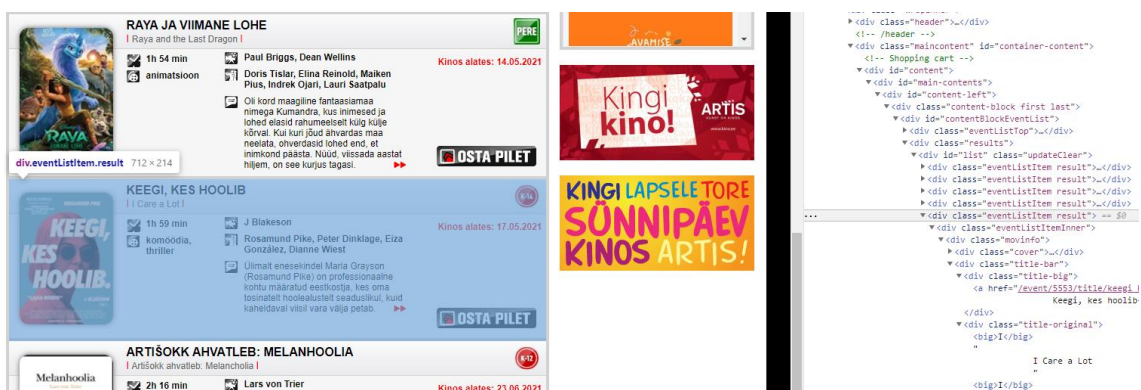
Projekti lahendamise käigus loodi kõigepealt programm, mis eraldab veebilehekülgedelt andmed. Selleks paigaldati PyCharm arenduskeskkond, loodi uus projekt ja seejärel paigaldati loodud projektile Scrapy raamistik. Antud programm kirjutati Python programmeerimiskeeles.

Selleks, et Artis Kino kodulehelt eraldada andmeid tuli luua *spider* Pythoni fail, kuhu kirjutati programm, mis saatis päringuid URL-dele, kust sai andmeid kinokavade kohta.

```
class ArtisSpider(scrapy.Spider):  
    name = 'artis'  
    base_url = 'https://www.kino.ee/'  
    start_urls = ['https://www.kino.ee/Movies/ComingSoon',  
                 'https://www.kino.ee/Movies/NowInTheatres']
```

Joonis 4. Andmete eraldamisel kasutatud URL-id

Vastavalt päringule leitud kino veebileht (Joonis 5).



The image shows a screenshot of the Artis Kino website interface on the left and its corresponding HTML structure on the right. The website displays movie listings for 'RAYA JA VIIMANE LOHE', 'KEEGI, KES HOOLIB', and 'ARTISOKK AHVATLEB: MELANHOOLIA'. The HTML structure on the right shows the DOM tree with various classes and IDs, including 'eventListitem.result', 'content-block', and 'eventListitem.result'.

Joonis 5. Kino kava HTML

Kui päring on õnnestunud, eraldatakse vajalikud andmed kasutades XPath'i (Joonis 6).

```
def parse_schedule(self, response):
    items = MovieSchedulesItem()
    schedule_type = \
response.xpath('//div[@id="contentBlockEventList"]//div[@class="contentArticleTab"]/h3/span/text()')[0]\
    .extract().strip()
    all_divs = response.xpath('//div[contains(@class, "eventListItem result")]')
    for m in all_divs:
        movie = m.xpath('.//div[@class="title-big"]/a/text()')[0].extract().strip()
        original_title = m.xpath('.//div[@class="title-original"]/text()')[1].extract().strip()
        date = m.xpath('.//div[@class="ajad"]/div/text()')[0].extract().split(':')[1].strip()
        genre = m.xpath('.//li[@class="genre"]/text()')[0].extract().strip()
        runtime = m.xpath('.//li[@class="aeg"]/text()')[0].extract().replace('\r\n', '').strip()
        movie_url = urljoin(self.base_url, m.xpath('.//div[@class="title-big"]/a/@href')[0].extract())

        items['cinema'] = 'Kino Artis'
        items['schedule_type'] = schedule_type
        items['movie'] = movie
        items['original_title'] = original_title
        items['date'] = date
        items['genre'] = genre
        items['runtime'] = runtime
        items['movie_url'] = movie_url

    yield items
```

Joonis 6. Andmete eraldamine XPath-iga

Seejärel seati üles MySQL andmebaas, kuhu edaspidi saadeti ja uuendati kinokava andmeid. Andmebaasi tegemiseks kasutati MySQL Workbench tööriista, mis on kasutajasõbralik graafiline liides.

Kõik andmed, mida Python programmiga sai eraldatud, saadeti MySQL andmebaasi (Joonis 5).

```

import mysql.connector

class MovieSchedulesPipeline:
    def __init__(self):
        self.create_connection()
        self.create_table()

    def create_connection(self):
        self.conn = mysql.connector.connect(
            host='localhost',
            user='root',
            passwd='123',
            database='movieschedules'
        )
        self.curr = self.conn.cursor()
        pass

```

Joonis 7. MySQL-ga ühenduse loomine

Pilt andmebaasi sisust (Joonis 8).

cinema	schedule_type	movie	original_title	date	genre	runtime	movie_url
Kino Artis	Varsti Kinos	Raya ja viimane lohe	Raya and the Last Dragon	23.04.2021	animatsioon	1h 54 min	<a href="https://www.kino.ee/event/5558/title/raya_ja_...">https://www.kino.ee/event/5558/title/raya_ja_...</a>
Kino Artis	Varsti Kinos	Aalto	Aalto	26.04.2021	dokumentaalfilm, eluloofilm	1h 43 min	<a href="https://www.kino.ee/event/5365/title/aalto/">https://www.kino.ee/event/5365/title/aalto/</a>
Kino Artis	Varsti Kinos	Isa	The Father	26.04.2021	draama	1h 37 min	<a href="https://www.kino.ee/event/5519/title/isa/">https://www.kino.ee/event/5519/title/isa/</a>
Kino Artis	Varsti Kinos	Nomaadimaa	Nomadland	26.04.2021	draama	1h 48 min	<a href="https://www.kino.ee/event/5522/title/nomaadi...">https://www.kino.ee/event/5522/title/nomaadi...</a>
Kino Artis	Varsti Kinos	Supernoova	Supernova	26.04.2021	draama	1h 35 min	<a href="https://www.kino.ee/event/5523/title/superno...">https://www.kino.ee/event/5523/title/superno...</a>
Kino Artis	Varsti Kinos	Pajutõotav noor naine	Promising Young Woman	26.04.2021	draama, Krimi	1h 53 min	<a href="https://www.kino.ee/event/5526/title/pajut%...">https://www.kino.ee/event/5526/title/pajut%...</a>
Kino Artis	Varsti Kinos	Õigluse ratsanikud	Retfærdighedens ryttere	26.04.2021	komöödia, draama	1h 57 min	<a href="https://www.kino.ee/event/5536/title/%C3%B...">https://www.kino.ee/event/5536/title/%C3%B...</a>

Joonis 8. Saadud andmed MySQL andmebaasis

Selleks, et Androidi rakendus saaks kuvada MySQL andmebaasis olevat informatsiooni, oli vaja üles seada veebiserver, mis suhtleks andmebaasiga. Antud lõputööks valiti Apache server ning kirjutati PHP skript, mis ühendas MySQL andmebaasiga, võttis tabelist andmed ning tagastas need JSON formaadis (Joonis 9).

```

<?php
$conn=mysqli_connect("localhost","root","123");
mysqli_select_db($conn,"movieschedules");

$qry="select * from artis_tb";

$row=mysqli_query($conn,$qry);

while($res=mysqli_fetch_array($row))
{
    $data[]=$res;
}
print(json_encode($data));
?>

```

Joonis 9. Andmebaasiga ühendamine ja tagastamine JSON formaadis

Androidi rakenduse arendamiseks valiti Android Studio arenduskeskkond ning programmeerimiskeeleks Java.

Selleks, et rakendus saaks ligipääsu internetti, lisati vastav pääsuluba *AndroidManifest.xml* faili (Joonis 10).

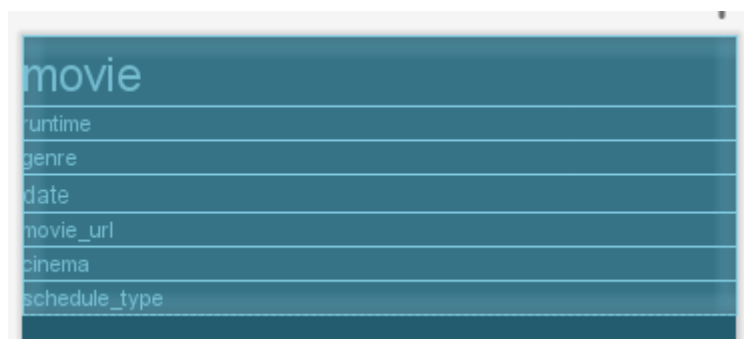
```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.csview">
    <uses-permission android:name="android.permission.INTERNET"/>

```

Joonis 10. Androidi pääsuluba

Andmete kuvamiseks loendina lisati *layout* kausta *row.xml*, kus kujundati loendis olevate ridade disaini (Joonis 11).



Joonis 11. Rea disain



Serveri kaudu andmete kättesaamiseks saatetakse päring serveri URL-le, kust loetakse sisse vastusena saadud andete voog ning tagastatakse saadud andmed (Joonis 12).

```
@Override
protected String doInBackground(String... strings) {
    try {
        URL url = new URL(strings[0]);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        BufferedReader br = new BufferedReader(
            new InputStreamReader(conn.getInputStream()));

        StringBuffer data = new StringBuffer();
        String line;

        while ((line = br.readLine()) != null) {
            data.append(line + "\n");
        }
        br.close();

        return data.toString();
    } catch (Exception ex) {
        return ex.getMessage();
    }
}
```

Joonis 12 Andmete lugemine serverist

Järgmisena loodi meetod, mis antud andmed saadab rakenduse *layout* kaustas olevasse *row.xml* faili, mida oli eelnevalt loodud, ning vastavalt muutujate nimedele määrati *TextView*-de *id* atribuudid (Joonis 13).

```

<TextView
    android:id="@+id/tv7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/colorPrimaryDark"
    android:text="runtime" />

<TextView
    android:id="@+id/tv6"
    android:layout_width="match_parent"
    android:layout_height="20dp"
    android:text="genre"
    android:textColor="@color/colorPrimaryDark"
    android:textSize="14sp" />

<TextView
    android:id="@+id/tv5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="@color/colorPrimaryDark"
    android:text="date"
    android:textSize="16sp" />

```

Joonis 13. *TextView* atribuudid

```

public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {
    LayoutInflater inflater = (LayoutInflater) getApplicationContext().getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    View row = inflater.inflate(R.layout.row, parent, attachToRoot: false);

    TextView tv5 = row.findViewById(R.id.tv5); //Date
    TextView tv6 = row.findViewById(R.id.tv6); //Genre
    TextView tv7 = row.findViewById(R.id.tv7); //Runtime
    tv5.setText(dt[position]);
    tv6.setText(gnre[position]);
    tv7.setText(rt[position]);
}

```

Joonis 14. Serverist saadud andmete määramine *TextView*-dele

Filmi kohta rohkema informatsiooni saamiseks lisati meetod, mille abil avaneb veebilehekülg vastavalt selle filmi kohta, millele oli vajutatud (Joonis 15).

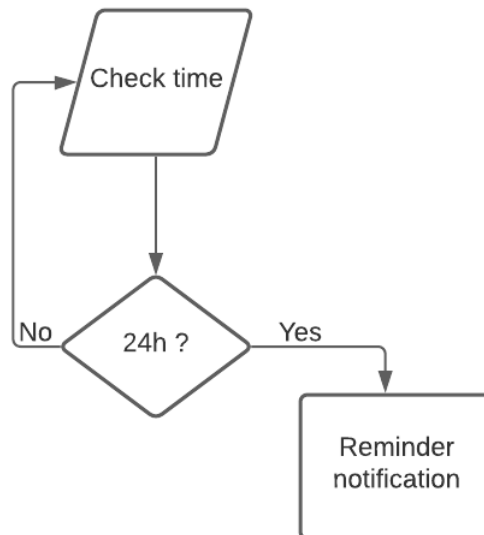
```

lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        String url = ((TextView) view.findViewById(R.id.tv8)).getText().toString();
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
    }
});

```

Joonis 15. Meetod veebi brauseri avamiseks

Selleks, et rakendus meeldetuletusi edastaks tuleb kontrollida filmi kuupäeva ja kellaega ning mobiiliseadme kella ja kuupäeva. Seejärel vaadatakse, kas nende aegade erinevus on rohkem kui 24 tundi. Kui on rohkem, siis jätkatakse aja kontrollimisega kui on vähem, siis edastatakse meeldetuletus(Joonis 16).



Joonis 16. Aja kontrollimise blokk skeem

## **6 Testimine**

Käesoleva diplomitöö kontekstis on testimise all mõeldud prototüübi moodultestimist, integratsioonitestimist ja süsteemtestimist.

### **6.1 Moodultestimine**

Moodultestimisel testiti veebikaabitsat, kus kontrolliti, kas programmi tagastatud andmed on korrektsed. Selleks sisestati veebilehe URL ning käivitati programm. Tagastatud andmed kuvati JSON vormis arenduskeskkonna terminali aknas. Seejärel andmete valideerimiseks võrreldi saadud JSON kinfo veebilehel oleva infoga vaadates, kas programmis tagastatud tekstiväljad on samasugused, mida kuvatakse veebilehel.

Androidi rakenduse moodultestimise käigus kontrolliti, et rakendus käivitub ning kuvab korrektset vaadet nii seadme emulaatoris kui ka mobiiliseadmes. Emulaatori puhul käivitati Androidi rakendus arenduskeskkonnas, mis avas virtuaalse Android seadme ning seadmes avati loodud rakendus. Rakenduse testimiseks mobiiliseadmes ühendati mobiiliseade arvutiga USB kaudu ning rakendus laeti mobiiliseadmesse ja vaadati üle, kas rakendus käivitub.

Moodultestimiste tulemused vastasid ootustele ehk veebist eraldatud andmed olid õiged ning Androidi rakendus käivitus edukalt.

### **6.2 Integratsioonitestimine**

Integratsioonitestimise käigus vaadeldi veebikaabitsa ja andmebaasi vahelist suhtlemist. Testimiseks võeti erinevad kinokava URL-d ja iga URL-i kohta tehti test, kus käivitati programm ning kontrolliti andmebaasis, kas programm sai edukalt andmebaasiga ühendatud ja sinna sisestatud andmed olid korrektsed.

Samuti testiti andmebaasi ja Androidi rakenduse vahelist suhtlemist, mille käigus vaadeldi kas andmebaasis ridade lisamine, väljade muutmine või rea kustutamine

kajastub ka Androidi rakenduse vaates. Antud testiks sisestati andmebaasi uus rida kasutajaliidese MySQL Workbench kaudu. Peale rea lisamist käivitati Androidi rakendus uuesti ning kontrolliti, kas uus rida on vaates kuvatud. Sarnaseid samme tehti ka andmebaasi väljade muutmise ning rea kustutamise puhul.

Integratsioonitestimiste tulemustena said õiged andmed sisestatud andmebaasi ning Androidi rakendus kuvas iga testi puhul korrektset informatsiooni.

### **6.3 Süsteemtestimne**

Süsteemtestimise puhul testiti mobiiliseadmes, kuidas rakendus toimib juhul, kui seadmes on andmeside välja lülitatud ja kui serveri masinas peatatakse server.

Seadmes käivitati rakendus, kus andmeside oli eelnevalt välja lülitatud. Tulemusena avanes rakendus, mis näitas tühja vaadet ning umbes 4 sekundit kuvati Java veateadet. Pärast seda kuvati jätkuvalt tühja vaadet.

Järgmisena vaadeldi, kuidas rakendus käitub kui server on peatatud ning rakendust üritatakse käivitada. Tulemusena rakendus avanes ja kuvati ainult tühja vaadet, aga ühtegi veateadet ei näidatud.

## 7 Kokkuvõte

Diplomitöö eesmärgiks oli luua rakendus, mis hõlbustaks kinokavade jälgimist ning kinokülastajate aja planeerimist. Lahenduse peamisteks nõueteks oli ajakohane kinokava kuvamine ning teavituse edastamine juhul kui kasutajat huvitav film on lisatud kavas.

Töö analüüsi käigus tehti ülevaade nii olemasolevatest mobiilirakendustest kui ka veebirakendustest, kust saab informatsiooni linastuvate filmide ja kinokavade kohta, kirjeldati projekti arenduskäiku ning testimiste tulemusi.

Töö lõpptulemuseks sai valmis rakenduse prototüüp, mis näitab ajakohast informatsiooni kinokavade kohta ja teavitab kui kasutajale huvi pakkuv film on lisatud kavas.

Valminud projekti täidab peamisi funktsionaalseid nõudeid, mis olid esialgu paika pandud ning tulevikus on võimalik rakendust edasi arendada ja lisada praktilisi funktsionaalsusi juurde. Üheks näiteks on lisada kalendri kuvamise funktsioon, mis seisneb selles, et kui oodatud film on lisatud kavas ning edastatakse teavitus, siis teavitusele lisaks näidatakse ka informatsiooni, mis on kasutaja kalendrisse märgitud samal kuupäeval. Teiseks on võimalik lisada pileti ostmise funktsiooni, mis võimaldab kohe läbi rakenduse osta pileti ning ei pea avama eraldi veebilehte. Kolmas lisafunktsionaalsus võib olla gruppide loomine, mis võimaldaks seltskonnaga kinokülastusi planeerida rakenduse kaudu. Samuti on tulevikus võimalik laiendada rakendust teistele kultuuriüritustele nagu näiteks teatrietendused või kontserdid.

## Kasutatud kirjandus

- [1] Apollo Cinema, "Apollo Kino," Apollo Cinema OÜ, 2017. [Online].  
<https://www.apollokino.ee/info/mobileapp>. [Aprill 2021].
- [2] Scrapy, "Scrapy 2.5 documentation," [Online].  
<https://docs.scrapy.org/en/latest/index.html>. [Aprill 2021].
- [3] Scrapy, "Architecture overview," [Online].  
<https://docs.scrapy.org/en/latest/topics/architecture.html>. [Aprill 2021].
- [4] Android Studio, "Save data using SQLite," [Online].  
<https://developer.android.com/training/data-storage/sqlite>. [Aprill 2021].
- [5] Python Software Foundation, "What is Python? Executive Summary," [Online].  
<https://www.python.org/doc/essays/blurb/>. [Aprill 2021].
- [6] Code Institute, "What is Java and why is it important?," [Online].  
<https://codeinstitute.net/blog/what-is-java/>. [Aprill 2021].
- [7] ProWebScraper, "The 5 Best Programming Languages for Web Scraping," [Online].  
<https://prowebscraper.com/blog/best-programming-language-for-web-scraping/>. [Aprill 2021].
- [8] Mozilla and individual contributors, "What is JavaScript?," [Online].  
[https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First\\_steps/What\\_is\\_JavaScript](https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript). [Aprill 2021].
- [9] TechTarget Contributor, "Ruby," [Online].  
<https://whatis.techtarget.com/definition/Ruby>. [Aprill 2021].
- [10] I. Blair, "14 Programming Languages for Mobile App Development," [Online].  
<https://buildfire.com/programming-languages-for-mobile-app-development/>. [Aprill 2021].
- [11] M. Heller, "What is Kotlin? The Java alternative explained," [Online].  
<https://www.infoworld.com/article/3224868/what-is-kotlin-the-java-alternative-explained.html>. [Aprill 2021].
- [12] Octopus Data Inc., "Easy Web Scraping for Anyone," Octopus Data Inc., 2020. [Online].  
<https://www.octoparse.com/>. [Aprill 2021].
- [13] Forum Cinemas, "Forum Cinemas," Forum Cinemas Eesti, [Online].  
<https://www.forumcinemas.ee/>. [Aprill 2021].
- [14] E. S., "SQLite vs MySQL – What's the Difference," 27 Märts 2019. [Online].  
<https://www.hostinger.com/tutorials/sqlite-vs-mysql-whats-the-difference/>. [Aprill 2021].
- [15] E. J., "MSQL Strengths and Weaknesses," Printer Friendly Format, [Online].  
<https://www.java-samples.com/showtutorial.php?tutorialid=929>. [Aprill 2021].

- [16] The PHP Group, "What is PHP?," The PHP Group, [Online].  
<https://www.php.net/manual/en/intro-what-is.php>. [April 2021].
- [17] ParseHub HQ, "Web Scraping for Developers," [Online].  
<https://www.parsehub.com/web-scraping-for-developers>. [April 2021].



## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Lisette Noor

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Rakendus kinokülastuse planeerimiseks", mille juhendaja on Meelis Antoi
  - 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.

29.04.2021

---

<sup>1</sup> 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 loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Rakenduse vaated

