

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Jelena Fjodorova 155542

**MITTEPROFESSIONAALIST  
TARKVARAARENDAJALE  
TÖÖKESKKONNA LOOMINE LOW-  
CODE/NO-CODE PLATVORMIL  
ALGKLASSIDE ÕPETAJA NÄITEL**

Bakalaureusetöö

Juhendaja: Mart Roost

Magistrikraad

Tallinn 2018

## **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: Jelena Fjodorova

21.05.2018

## **Annotatsioon**

Käesoleva lõputöö „Mitteprofessionaalset tarkvaraarendajale töökeskkonna loomine low-code/no-code platvormil algklasside õpetaja näitel“ eesmärgiks on "muuta" konkreetne kooliõpetaja mitteprofessionaalset tarkvaraarendajaks, et tema saaks low-code/no-code platvormi tööriistu kasutades luua endale ja oma õpilastele vajalikku tarkvararakendust.

Eesmärgi saavutamiseks uuriti paljusid low-code/no-code platvorme ja nende tööriistu. Samuti õpetati õpetajat kui arendajat platvormi kasutama.

Töö tulemusena valmis mobiilirakendus reeglite õppimiseks algklassi õpilaste jaoks.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 39 leheküljel, 8 peatükki, 22 joonist, 5 tabelit.

## **Abstract**

### **Creating a working environment for a nonprofessional software developer on the low-code/no-code platform by example of an elementary school teacher**

The aim of the thesis «Creating a working environment for a nonprofessional software developer on the low-code/no-code platform by example of an elementary school teacher» is to «transform» a school teacher into a nonprofessional software developer, so that she could create a software application for herself and her students using the low-code/no-code platform tools.

To achieve the aim, many low-code/no-code platforms and their tools was tested. Also the teacher as the developer was taught how to work on the platform.

The result of this work is a mobile application for learning rules for primary school students.

The thesis is in Estonian and contains 39 pages of text, 8 chapters, 22 figures, 5 tables.

## Lühendite ja mõistete sõnastik

OS	<i>Operation system</i> , operatsioonsüsteem
EUD	<i>End-User Development</i> , lõppkasutajate arendamine
EUP	<i>End-User Programming</i> , lõppkasutajate programmeerimine
CMS	<i>Content Management System</i> , sisuhaldustarkvara
MVC	<i>Model-view-controller</i>
HTML	<i>Hypertext Markup Language</i> , hüperteksti märgistuskeel

## Sisukord

1 Sissejuhatus.....	10
1.1 Probleemi ja tausta kirjeldus.....	10
1.2 Ülesande püstitus .....	11
1.3 Metoodika .....	11
1.4 Ülevaade tööst.....	11
2 Ettevõtte tutvustus.....	12
2.1 Sillamäe Kannuka Kool .....	12
2.2 Algkooli õppekava .....	12
2.3 Õpetamismetoodikad .....	12
2.4 Algkooli õppematerjalid .....	13
2.5 Algklassi õpetaja põhiülesanded.....	14
3 Tarkvara arendamine .....	15
3.1 IT-maailma probleem.....	15
3.2 Tarkvara realiseerimisevahendid .....	15
3.3 Mobiilirakendus .....	16
4 End-user development .....	21
4.1 Definiitsioon.....	21
4.2 Lõppkasutajate programmeerimine .....	23
4.2.1 Visuaalne programmeerimine .....	23
4.3 Meta-design.....	23
5 Low-code/no-code .....	24
5.1 Low-code/no-code platvormid.....	24
5.2 Low-code mobiilarenduse realiseerimiseks.....	25
5.3 No-code mobiilarenduse realiseerimiseks .....	27
5.3.1 No-code platvormide võrdlus .....	28
6 Mobiilirakenduse realiseerimine.....	31
6.1 Sobilik tarkvara.....	31

6.1.1 Mobincube platvormi kasutajamugavus .....	31
6.1.2 Mobincube platvormi võimalused .....	31
6.2 Mobiilirakenduse kavandamine .....	32
6.3 Realiseerimise etapid .....	36
6.4 Õpetamine .....	40
6.5 Integreerimine .....	40
7 Analüüs .....	42
7.1 Loodud rakenduse analüüs.....	42
7.2 SWOT analüüs .....	43
7.3 Rakenduse hindamine (küsitlus).....	44
7.4 Rakenduse valideerimine .....	46
7.5 Edasine areng.....	46
7.6 Rakenduse installeerimine .....	47
8 Kokkuvõte.....	48
Kasutatud kirjandus .....	49
Lisa 1 - Küsitlus .....	51
Lisa 2 – Küsitluse vastused.....	53
Lisa 3 – Kannuka Kooli direktori arvamus.....	55
Lisa 4 – Mobiilirakenduse ekraanipildid .....	56

## Jooniste loetelu

Joonis 1. Õpetaja ülesanded ja töötulemid.....	14
Joonis 2. Mobiilsete operatsioonsüsteemide turuosa maailmas seisuga veebruar 2018.	17
Joonis 3. Google Play statistika: saadavate rakenduste arvu kasv detsembrist 2009 märtsini 2017 [10].....	18
Joonis 4. AppStore statistika: saadavate rakenduste arvu kasv juunist 2008 jaanuarini 2017 [10].....	18
Joonis 5. Google Play ja AppStore rakenduste laadimise arv aastatel 2014-2016 [10].	19
Joonis 6. Tarkvara arendamise protsess professionaalse arendaja juhul. ....	20
Joonis 7. Prognoositud USA töökohtade suhtarv aastal 2012 [12].....	22
Joonis 8. Low-code platvormid mobiilarenduseks 2017. aasta alguses [20].....	26
Joonis 9. Mobiilrakenduse realiseerimine low-code platvormi abil. ....	27
Joonis 10. Mobiilrakenduse arendamine no-code platvormi abil. ....	28
Joonis 11. Domeenimudel.....	33
Joonis 12. Õpetaja eesmärgid platvormi kasutamisel.....	34
Joonis 13. Õpilase eesmärgid rakenduse kasutamisel.....	35
Joonis 14. Õpetaja “muutmine” tarkvaraarendajaks.....	37
Joonis 15. Õpetaja kasutajaeesmärgid Mobincube platvormis. ....	39
Joonis 16. QR-kood mobiilrakenduse MyStudy installeerimiseks.....	47
Joonis 17. Õpilaste vastused küsitlusele. ....	53
Joonis 18. Õpetajate vastused küsitlusele. ....	54
Joonis 19. Kannuka Kooli direktori arvamus loodud rakenduse kohta. ....	55
Joonis 20. MyStudy rakenduse pealehelülg.....	56
Joonis 21. MyStudy rakenduse eesti keele lehekülg.....	56
Joonis 22. MyStudy rakenduse “Arvsõnad” teema reeglite lehekülg.....	57



## **Tabelite loetelu**

Tabel 1. No-code platvormide omaduste võrdlemine [21]. .....	28
Tabel 2. Mobiilirakenduse loomise tähtaegade võrdlemine. ....	42
Tabel 3. Mobiilirakenduse kasutamise SWOT analüüs. ....	43
Tabel 4. Õpilaste vastused küsitlusele mobiilirakenduse kasutamise kohta. ....	44
Tabel 5. Õpetajate vastused küsitlusele mobiilirakenduse kasutamise kohta. ....	45

# 1 Sissejuhatus

Antud lõputöö teema «Mitteprofessionaalset tarkvaraarendajale töökeskkonna loomine low-code/no-code platvormil algklasside õpetaja näitel» tuleneb vajadusest leida sobilik tarkvara, mis võimaldaks mitteprofessionaalset tarkvaraarendajal ehk algklasside õpetajal luua iseseisvalt või koostöös professionaalse tarkvaraarendajaga vajalik töökeskkond oma õpilastele. Teema on aktuaalne, kuna praegu on professionaalsete koodi tasemel tarkvaraarendajate defitsiit ning nende teenused maksavad väga palju.

## 1.1 Probleemi ja tausta kirjeldus

Kaasaegne inimene peab hästi tundma info- ja kommunikatsioonitehnoloogiaid, kuna need on lahutamatu osa meie elust. Üha rohkem integreeritakse tarkvaraarendusi koolidesse, sealhulgas algklassidesse. Seoses aktiivse IT-lahenduste kasutusele võtmisega õppimise sfääris muutub ka õpetamismetoodika. Arvutid, arvutitahvlid, nutitelefonid ja teised seadmed on tööriistad, mis võimaldavad meid kiiresti töödelda informatsiooni. Sellised tööriistad võivad olla tugevaks õpetamise ning kommunikatsiooni vahendiks õpetaja, õpilaste ja vanemate vahel.

Koolides tarkvaraarenduste kasutamise küsimuse kõrval seisneb küsimus nende kättesaadavusest. Kui infotehnoloogiate kasutamine praegu dünaamiliselt areneb, siis inimesi, kes seda arendaksid on vähe. Pole saladus, et maailmas on professionaalsete koodi tasemel tarkvaraarendajate defitsiit. Lisaks sellele, kvalifitseeritud tarkvaraarendaja teenused on liiga kallid. Arvestades seda, et tavalisel algklassiõpetajal on keskmiselt seitse põhiainet, võib professionaalsete teenuste kasutamine tema jaoks olla liiga kallis ning ebasoodne.

Selle probleemi lahenduseks on niinimetatud low-code/no-code platvormid, mis on suunatud just mitteprofessionaalsete tarkvaraarendajatele ning võimaldavad mitte-IT inimestel luua vajalikke tarkvararakendusi.

## 1.2 Ülesande püstitus

Käesoleva lõputöö põhieesmärgiks on muuta konkreetne kooliõpetaja mitteprofessionaalist tarkvaraarendajaks nii, et kasutades low-code/no-code platvormi tööriistu saaks ta luua oma õpilastele vajalikke lihtsamat tüüpi tarkvararakendusi.

## 1.3 Metoodika

Lähtudes ülesande püstitusest tulenevad järgmised töö metoodika sammud:

- Õpetaja töövaldkondade ja tarkvara nõuete analüüsimine
- Erinevate low-code/no-code arendusplatvormide ülevaade, võrdlemine ning sobivama platvormi väljavalimine
- Õpetaja tarkvaraarenduskeskkonna kavandamine ja loomine
- Õpetaja koolitamine, juhendamine ja toetamine
- Tulemuste testimine, hindamine ja analüüs

## 1.4 Ülevaade tööst

Töö koosneb kuuest osast. Järgnevas teises osas defineeritakse lühidalt algkooli olemus, tutvustatakse algkoolis rakendatavaid õppimismetoodikaid ning õppematerjale, antakse ülevaade õpetaja peamistest tööülesannetest. Kolmandas töö osas uuritakse hetkeolukorda IT-maailmas ning tarkvaraarenduses, üldistatakse ja täpsustatakse töös lahendatavat probleemi. Neljandas osas kirjutatakse lõppkasutajate poolt tehtavast arendamisest (end-user development), mis on teoreetiliseks aluseks low-code/no-code platvormidel tarkvaraarendusele ning lahenduseks eelmises osas väljatoodud põhiprobleemile. Seejärel viiendas osas kirjutab autor low-code/no-code platvormidest, võrdleb neid ning valib neist sobivaima mobiilirakenduse arendamiseks. Töö kuuendas osas kirjeldatakse mobiilirakenduse loomise protsessi valitud no-code platvormil ning käsitletakse platvormi kasutajamugavust, realiseerimise etappe, tähtaegu. Samuti kirjeldatakse mitteprofessionaalist tarkvaraarendaja õpetamise ja tema loodud rakenduste integratsiooni võimalust. Lõpuks analüüsitakse töö tulemusi.

## **2 Ettevõtte tutvustus**

Alljärgnevas peatükis kirjutatakse täpsemalt algkooli keskkonnast. Kirjeldatakse algkoolis kasutatavaid õppematerjale ning räägitakse erinevatest õpetamismetoodikast. Käesolevas peatükis kõige rohkem pöörab autor tähelepanu infotehnoloogiate kasutamisele algkoolis.

Kooliks on valitud konkreetne põhikool Sillamäe linnas – Sillamäe Kannuka Kool.

### **2.1 Sillamäe Kannuka Kool**

Algkool on laste üldharidusasutus, mis pakub algharidust peamiselt emakeele, võõrkeele, matemaatika, looduse, inimeseõpetuse ning kehalise kasvatuse valdkonnas. Samuti algkool on laste esimene ja kohustuslik üldhariduse etapp. Algkoolis saab õppida alates 6-7 aastast ning algkoolis õppimine Eestis kestab 4 aastat. Vastavalt Eesti Vabariigi põhikooli- ja gümnaasiumiseadusele [1], klassi täitumise ülemine piirnorm peab olema 24 õpilast.

Kannuka Kool on alustanud oma tegevust 10. septembril 1990.a Sillamäe linnas. Kannuka Kool on vene õppekeelega munitsipaalõppeasutus, mis pakub alg- ja põhiharidust. Alates 2005. aastast tegeleb kool eestikeelse aineõpetusega 1.-9. klassides [2].

### **2.2 Algkooli õppekava**

Sillamäe Kannuka Koolis kehtib põhikooli riiklik õppekava. Kooli õppekava on esitatud Kannuka Kooli ametlikul veebilehel [3].

### **2.3 Õpetamismetoodikad**

Õppimine ja õpetamine on peaaegu sama vanad kui inimkond. Koos inimese arenguga on muutunud ka õppimise ja õpetamise olemus. Iga aastaga tekkivad juurde uued võimalused ja meetodid õppimiseks ja õpetamiseks. Muidugi õppimine on kahepoolne protsess, mis eeldab nii õpetaja kui ka õpilase jõupingutusi, et saavutada sobivat tulemust. Õpetaja

saab edasi anda rohkem teadmisi ainult siis, kui tema õpetamine põhineb teaduslikult tõestatud ja palju kordi uuritud meetoditel [4].

Kõige sagedamini klassifitseeritakse meetodeid teadmiste saamise allikate järgi. Vastavalt sellele lähenemisele võib meetodeid jagada kolmeks gruppiks:

1. Verbaalsed meetodid: lugu, seletus, vestlus, töö õpiku või raamatuga.
2. Visuaalsed meetodid: vaatlus, filmid, näitvahendite tutvustamine.
3. Praktilised meetodid: suulised ja kirjalikud harjutused, graafilised ning laboritööd.

Iga õpetaja valib oma meetodid õpetamiseks, kuid nendest kolmest grupist on võimalik valida ühte, mis on kõige efektiivsem õppematerjali omandamiseks ja see on praktiline. Tehes praktilisi ülesandeid saab õpilane rohkem teadmisi, arendab oma nägemis- ja lihasmälu ning saab emotsioone. Ühendades need kolm meetodeid on võimalik luua õppimiskeskond, mis võiks olla meeldiv ja kasulik nii õpetajale kui ka õpilastele.

## 2.4 Algekooli õppematerjalid

Eelmises peatükis valis autor praktilise õppimismeetodi kui kõige efektiivsema meetodi õppimiseks. Praktilisi ülesandeid saab teha nii kirjalikult vihikus või töövihikus, kui ka elektroonselt. Elektroonseks ülesande lahendamiseks on mõeldud erinevad keskkonnad, mida iga aastaga hakatakse üha rohkem koolidesse integreerima. Järgnevalt on toodud kõik õppe- ja õpetamismaterjalid, mis kasutatakse Kannuka Kooli algklasside tundides.

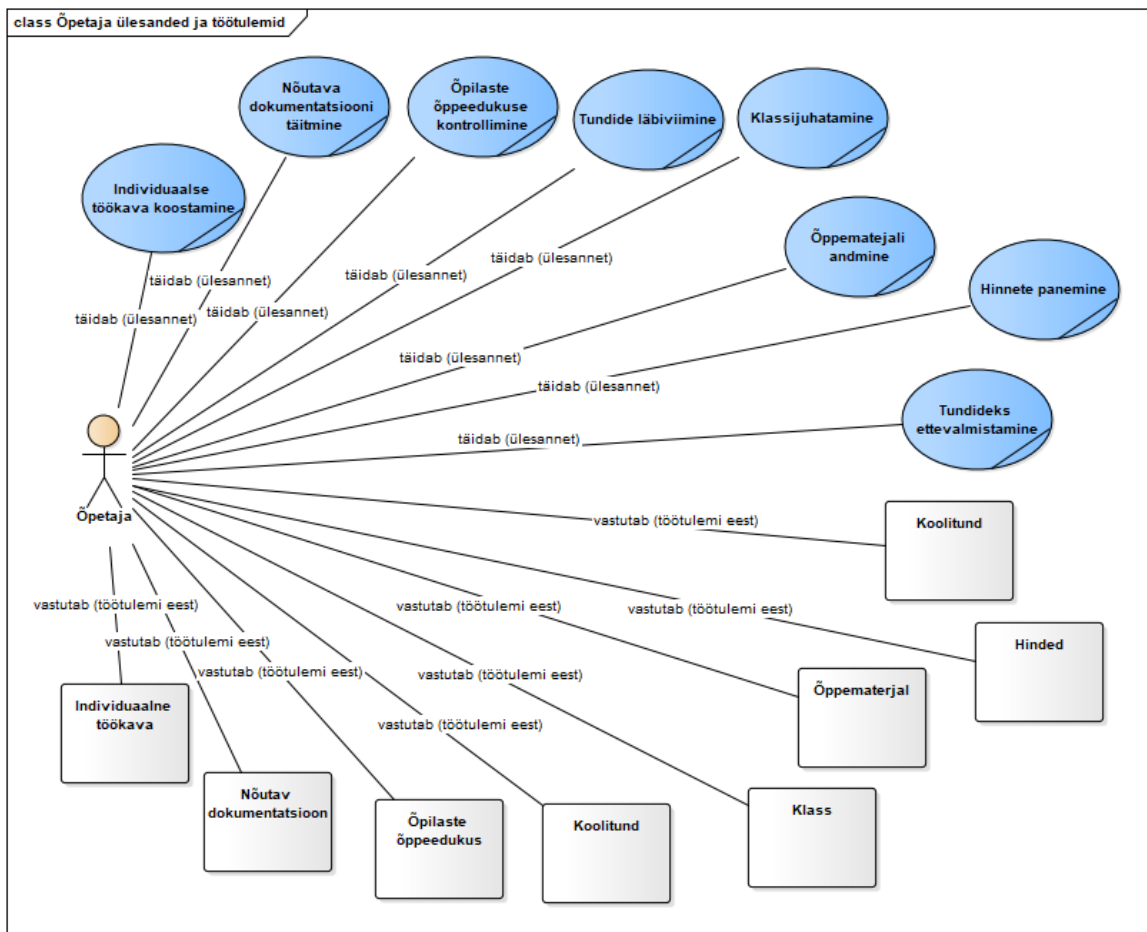
- Kirjavarasid
  - Õpikud
  - Töövihikud
  - Õpetaja poolt koostatud töölehed
  - Raamatud
- E-materjalid
  - e-õpikud (matemaatika, vene keel)
  - [quizizz.com](http://quizizz.com)
  - [kahoot.it](http://kahoot.it)
  - [learningapps.org](http://learningapps.org)
  - [plickers.com](http://plickers.com)

On näha, et algkoolis kasutatakse üsna tihedalt erinevaid e-keskkondi. Need keskkonnad võimaldavad õpilastel lahendada õpetaja poolt koostatud teste või ülesande, lugeda materjali või mängida, omandades niimoodi mingi aine materjali läbi mängu.

Infotehnoloogiad lihtsustavad õppimisprotsessi, säästavad aega ning arendavad õpilastel vajalike oskusi tulevikuks.

## 2.5 Alklassi õpetaja põhiülesanded

Edasi on toodud autori poolt tehtud diagramm, mis kirjeldab alklasside õpetaja põhilisi tööülesandeid. Tööülesanded on esitatud business use case-idenana ning vastutatavad töötulemid klassidena.



Joonis 1. Õpetaja ülesanded ja töötulemid.

## **3 Tarkvara arendamine**

Käesolevas peatükis kirjutab autor tarkvaratoodete loomise kohta. Selle töö osa eesmärgiks on määratleda antud töö fookuses olev IT-maailma probleem. Samuti kirjeldab autor professionaalsete programmeerijate tarkvarade loomise viise. Kuna antud töö lõpus luuakse mobiilirakendust, siis keskendub autor mobiilirakenduse arendamise teemale. Kirjutatakse mobiilirakenduse populaarsusest, operatsioonisüsteemidest ning rakeduste laadimise keskkondadest.

### **3.1 IT-maailma probleem**

Erialakirjanduses [5] räägitakse professionaalsete koodi tasemel tarkvaraarendajate defitsiidist. Praegu on üsna raske häid programmeerijaid leida ning nende teenused maksavad väga palju. Rääkides Eestist võib märgata, et hinnad IT-teenuste eest on juba läinud välismaa hindadeks. Uptime'i tehnoloogiajuht Raimo Seero oma intervjuus Äripäev ajakirjale rääkis: "Mida aeg edasi, seda paremini müüvad Eesti IT-ettevõtted oma teenust välismaale ning seda vähem ja seda kallimalt saab seda Eesti turul osta,». «Põhimure on see, et mina mõtlen arendaja väljamüügis Norra, Saksa, Šveitsi turu hindadele, mis ei ole see teenuse taseme hind, mida eestlane on harjunud maksma. Ma lihtsalt näen, et see tendents läheb ajapikku hullemaks,» lisas Seero [6].

Seoses sellega, et professionaalseid tarkvaraarendajaid on praegu vähe, maksavad nende teenused väga palju. Näiteks kui rääkida kodulehe valmistamisest, siis hinnad Eestis varieeruvad vahemikus 400 kuni 6000 eurot (hind sõltub kodulehe raskusest).

Kvalifitseeritud tarkvaraarendajate defitsiit ning nende teenuste kõrged hinnad sunnivad ettevõtteid või ärikasutajaid otsima alternatiivseid võimalusi tarkvara arendamiseks.

### **3.2 Tarkvara realiseerimisevahendid**

Arendaja poolt valitud programmeerimiskeel võib olla üks kõige olulisem aspekt ning see on esimene samm tarkvaraarenduse realiseerimiseks. See on tähtis, kuna valitud keel võib sõltuda kasutatavast operatsioonisüsteemist, kuna iga süsteem töötab paremini mingite

konkreetsete programmeerimiskeeltega. Inimene, kes ei omandanud IT-eriala ning tahab saada professionaalseks tarkvaraarendajaks, peab maksma programmeerimiskeelte kursuste eest, mis nõuab suure summa kulutamist. Samuti inimene, kellel on soov saada professionaalseks tarkvaraarendajaks peab mõistma, et hea programmeerija tavaliselt teab rohkem kui ühte programmeerimiskeelt. Kõige populaarsemad keeled on praegu Java, C++, C, Python, PHP, JavaScript, SQL. IT Koolitus firma hinnakirja kohaselt näiteks ühe programmeerimiskeele kursus maksab 129 eurot [7]. BCS Koolitus pakub Java, Python, programmeerimise kursust 499 euro eest, JavaScript'i kursust 399 euro eest [8]. Mitte kõik inimesed ei suuda maksta sarnast summat.

Professionaalse koodi tasemel tarkvaraarendajale pakutakse tänapäeval üsna laia valikut tarkvara arenduskeskkondi. Paljud neist on tasuta, paljud tasulised. Arvestades seda, et kvalifitseeritud programmeerija teab rohkem kui ühte programmeerimiskeelt, siis peab ta omandama mitu arenduskeskkonda korraga, kuna erinevad programmeerimise eesmärgid nõuavad erinevaid keskkondi arendamiseks.

Kõik sellised aspektid takistavad mitte IT-inimesi programmeerijateks saamast ning sunnivad neid alternatiivseid tarkvaraarendamise viise otsima.

### **3.3 Mobiilirakendus**

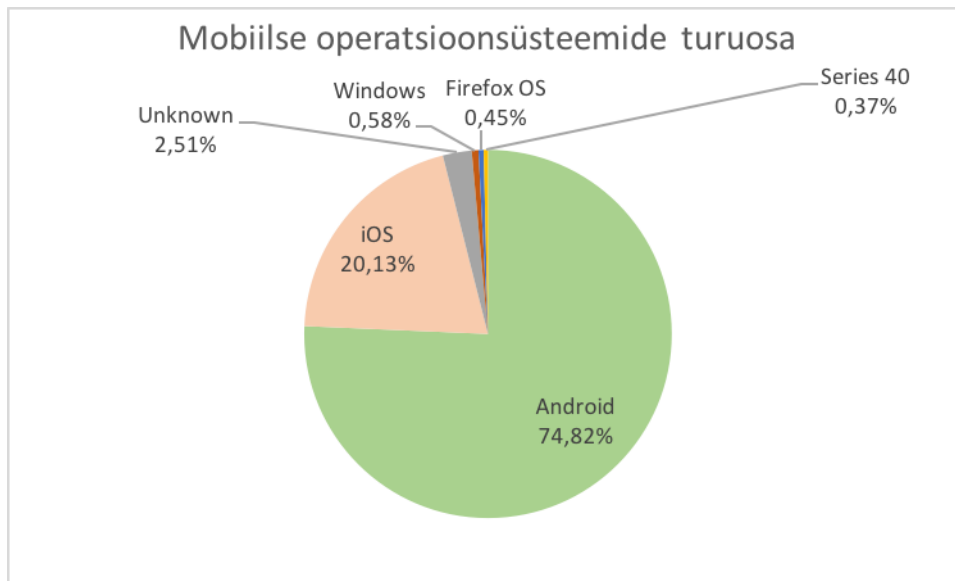
Antud töö lõpp-produktiks on mobiilirakendus algklasside jaoks, seega edasi on täpsemalt kirjeldatud mobiilirakenduse olemus.

Mobiilirakendus on nutitelefonidele, tahvelarvutitele ja muudele mobiilseadmetele mõeldud tarkvara [25].

Mobiilirakenduste operatsioonsüsteemideks on Windows Mobile, Symbian, RIM, Android, iOS ja teised. Kõige populaarsemad nendest tänapäeval on Android ja iOS.

Vastavalt [gs.statcounter.com](http://gs.statcounter.com) veebilehe statistikale [9] on koostatud järgmine diagramm, mis näitab mobiilse operatsioonsüsteemide turuosa maailmas seisuga veebruar 2018.

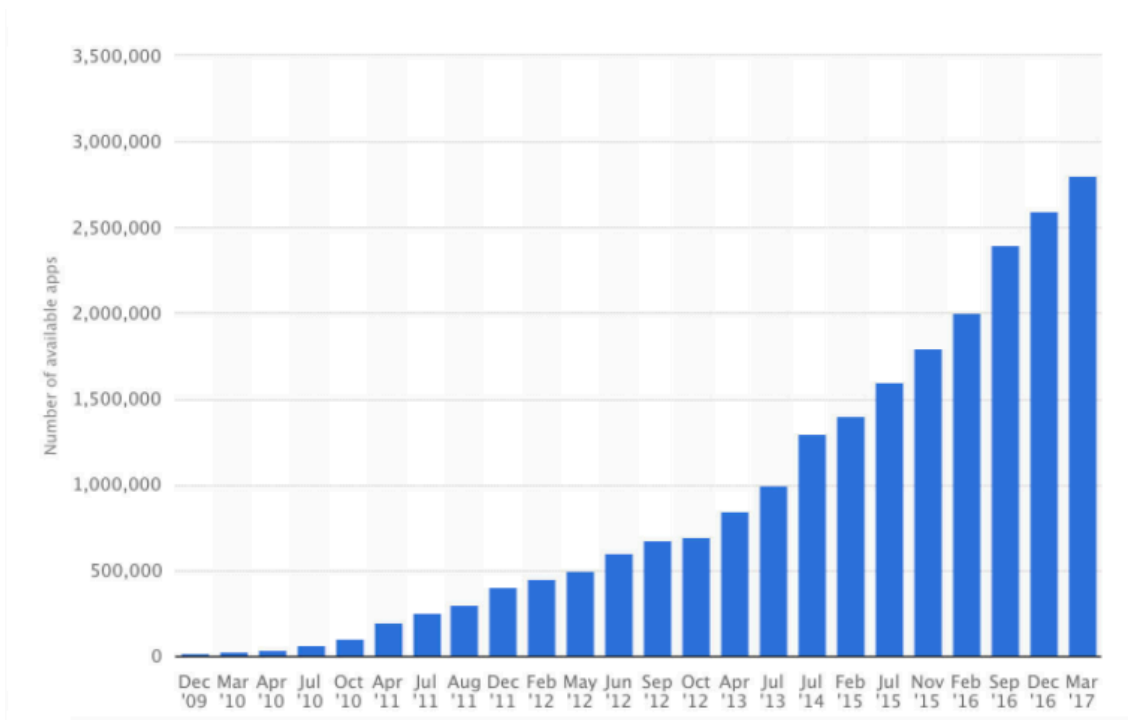




Joonis 2. Mobiilsete operatsioonsüsteemide turuosasa maailmas seisuga veebruar 2018.

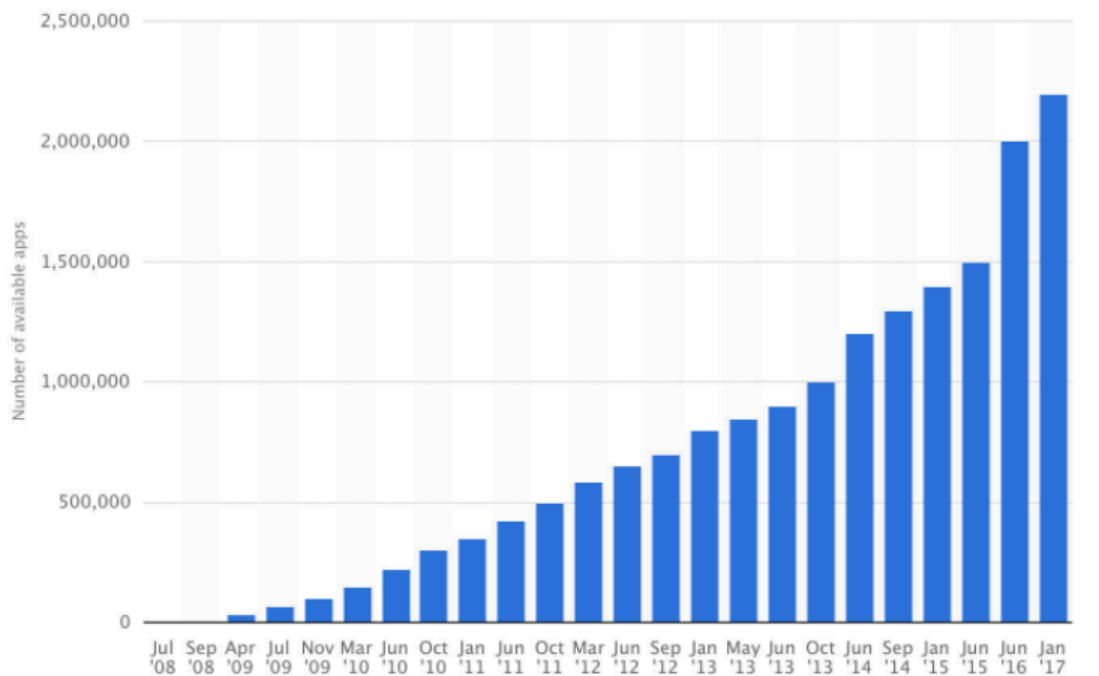
Diagrammist on näha, et kõige levinum operatsioonsüsteem, mida kasutavad inimesed oma nutitelefonides on Android, mis moodustab 74,82%. Teisel kohal on Apple'i toodete operatsioonsüsteem iOS, mille turuosasa protsent on 20,13%. 2,51% on tundmatul (Unknown) operatsioonsüsteemil. Firefox OS ja Series 40 süsteemidel on kõige madalam protsent ehk 0,45% ja 0,37%.

Järgmiselt kirjeldatakse täpsemalt kõige populaarsemaid operatsioonsüsteeme Android ja iOS. Apple ja Google on kaks tugevad tootjad turul. Need pakuvad meile nii integreeritud rakendusi kui ka kolmanda osapoole rakendusi. Selliste "võõraste" mobiilirakendusi paigutuseks on rakenduste poed Google Play (Google, Android) ja AppStore (Apple, iOS). Mõlemad oli loodud 2008. aastal. Jaanuaris 2017 oli Google Play rakenduse poes juba 2,7 miljonit ning AppStore'is 2,2 miljonit rakendusi [10]. Sellised näitajad on esitatud järgmistel diagrammidel, mis kirjeldavad saadavate rakenduste arvu kasvu statistikat Google Play Store'is ja AppStore'is kindlas ajavahemikus.



Joonis 3. Google Play statistika: saadavate rakenduste arvu kasv detsembrist 2009 märtsini 2017 [10].

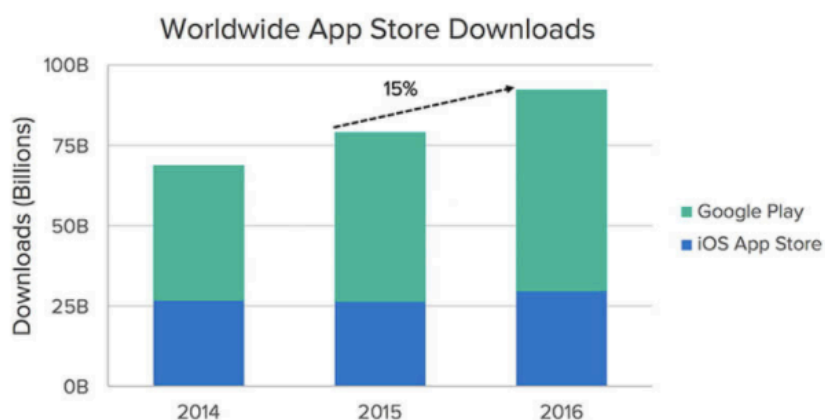
Diagrammist on näha, et saadavate rakenduste arv Google Play rakenduste poes on dünaamiliselt kasvanud vahemikus 2009-2017. Kui poodi ilmumise alguses oli ainult paar tuhat rakendust, siis eelmise aasta alguses näitaja ületas 2,7 miljonit rakendust.



Joonis 4. AppStore statistika: saadavate rakenduste arvu kasv juunist 2008 jaanuarini 2017 [10].

Ülaltoodud diagrammist on näha, et AppStore rakenduste poes aastatel 2008-2017 oli sujuv rakenduste kasv. Kui 2008. aastal oli AppStore'is 500 saadavaid rakendusi, siis eelmise aasta jaanuaris nende hulk jõudis 2,2 miljonini.

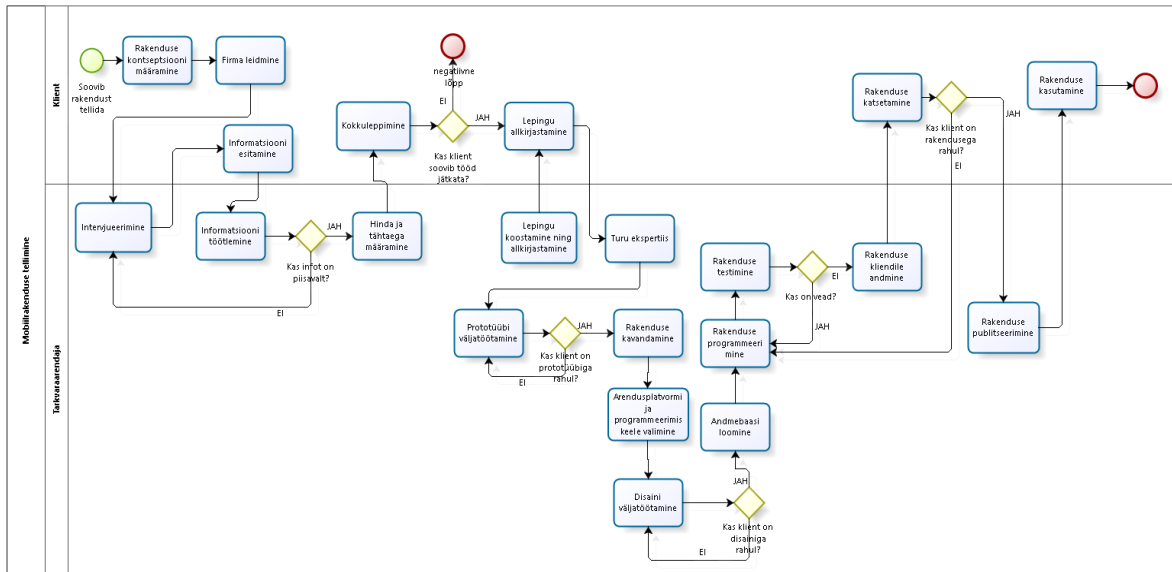
Maailma juhtiva mobiilirakenduste turu andmete tarnija App Annie statistika näitab, et Google Play rakenduse poest laadimise arv 2016. aastal jõudis 75 miljardini, mis on 15% võrra suurem kui 2015. aastal ning laadimise arv on paar korda suurem kui AppStore'is, kust oli laaditud umbes 25 miljardid rakendusi. Need andmed on esitatud järgmisel tulpdiaagrammil.



Joonis 5. Google Play ja AppStore rakenduste laadimise arv aastatel 2014-2016 [10].

Statistika andmetest on näha, et Android operatsioonsüsteem on kõige levinum tänapäeval ning sellest järeldub, et kõige rohkem laaditakse rakendusi Google Play poest. Rakenduste realiseerimine on üsna keerukas ja kallis protsess. Disaini ja back-end'i väljatöötamine nõuab väga palju aega ja pingutusi, seega professionaalse tarkvaraarendaja teenused maksavad väga palju.

Edasi on toodud antud töö autori poolt tehtud Bizagi diagramm, mis näitab autori arusaamist sellest, kuidas loovad kliendile mobiilirakendust professionaalse koodi tasemel tarkvaraarendajad. Protsess koosneb tegevustest. Iga tegevus peab olema täidetud kas kasutaja või kliendi poolt.



Joonis 6. Tarkvara arendamise protsess professionaalse arendaja juhul.

Antud diagrammil on esitatud tarkvaraarenduse arendusprotsessi etapid professionaalse tarkvaraarendaja ja kliendi vahel. On näha, et tarkvaraarendaja peab kogu töö käigus arvestama kõiki kliendi soove ja nõudeid ning probleemide tekkimise puhul neid kohe lahendada. Just seetõttu arendusprotsess on üsna mahukas ja kallis ning võtab palju aega.

## 4 End-user development

Tänapäeval arvutikasutamine on väga laialt levinud ning hõlmab enamus ameteid. Nende hulka kuuluvad juhid, insenerid, raamatupidajad, õpetajad, arstid, teadlased, kindlustusandjad, müügimehed ja paljud teised. Nende tarkvara vajadused on mitmekesised, keerukad ja sageli muutuvad. Professionaalsed tarkvaraarendajad ei suuda otseselt kõiki neid vajadusi täita oma piiratud valdkonna teadmiste tõttu. Juba 2011. aastal USA Töö- ja Statistikaamet prognoosis, et 2012. aastaks peaks Ameerika Ühendriikides olema vähem kui 3 miljonit professionaalset tarkvaraarendajat ning rohkem kui 55 miljonit töötajat, kes kasutavad arvutustabeleid ja andmebaase oma töö jaoks. Samal aastal näitas Gartner aruannet, et 2014. aastaks loovad mitteprofessionaalsed arendajad vähemalt 25% ärirakendustest. Sellest järeldub, et mitteprofessionaalset tarkvaraarendajal peab olema võimalus osaleda tarkvara arendusprotsessis ilma märkimisväärsema programmeerimiskeele oskuseta [13].

Selleks, et lahendada professionaalsete tarkvaraarendajate defitsiidi probleemi ning anda võimalust mitte IT-sfäärist inimestele osaleda tarkvara arendusprotsessis on mõeldud lõppkasutajate arendus ehk end-user development (EUD). Sellest kirjeldatakse antud peatükis.

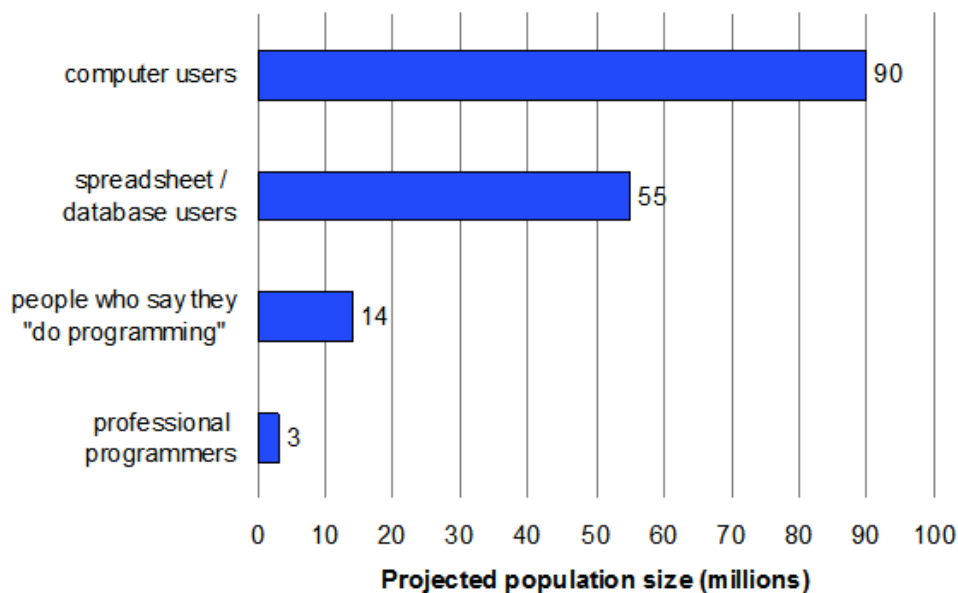
### 4.1 Definiitsioon

Lõppkasutaja arendust saab määratleda kui meetodeid, tehnikaid ning tööriistu, mis võimaldavad tarkvarasüsteemide kasutajatel, kes ei ole professionaalsed tarkvaraarendajad, mingis etapis luua, muuda või laiendada tarkvara [14].

Lõppkasutajad (end-users) on viimased tarkvaraarenduse kasutajad. Lõppkasutajate poolne (ise)arendamine võimaldab inimestel, kellel ei ole kõrgel tasemel programmeerimise oskust, tekitada automaatset käitumist ja luua erinevaid keerulisi andmeobjekte. See omakorda aitab vähendada erinevusi kasutajate ja arendajate vahel [15].

EUD võimaldab lõppkasutajatel kasutajaliidest ja tarkvara funktsionaalsust arendada ja kohandada. Seega on lõppkasutaja arendus väga väärtuslik, kuna lõppkasutajad teavad oma konteksti ja vajadusi paremini kui keegi teine, isegi professionaalne

tarkvaraarendaja. EUD rakendamine tarkvara arendusprotsessis laiendab tarkvara arendustegevuse ulatust, võimaldades osaleda protsessis laiemale inimeste ringile.



Joonis 7. Prognositud USA töökohtade suhtarv aastal 2012 [12].

Joonisel on esitatud USA's erinevates töö kategooriates töökohtade suhe 2012. aastal. Andmed põhinevad Ameerika Ühendriikide föderatiivsel statistikal. Joonisel on näha, et arvutikasutajad moodustavad kõige suurema osa – 90%, edasi lähevad arvutustabelite ja andmebaaside kasutajad – 55%. Inimesed, kes ütlevad, et nad “tegelevad programmeerimisega” moodustavad 14% ning professionaalseid programmeerijaid on kõige vähem – 3%. Andmetest järeldub, et inimesi, kes vajavad tarkvara kasutamist oma töös on palju aga inimesi, kes seda arendaksid on vähe.

EUD meetodika annab võimalust kaasata tarkvaratoote arendusprotsessi töötajaid erinevatest valdkondadest, kellel on selge arusaam sellest, mida selle toote kasutaja vajab. Konkreetse eriala töötaja on oma valdkonnas kitsamalt suunatud ning võib olla kasulik tootearenduse jaoks, seega meelitades rohkem kasutajaid. Tänu programmeerimisele lõppkasutaja osalemisega muutub tarkvaraarenduse olulisimaks teguriks tähelepanu pööramine kvaliteedile.

Seoses sellega, et kasutajad suhtlevad suure hulga kolleegidega, võib tarkvara kood muutuda teistele nähtavaks ja seetõttu vastuvõtlikumaks rünnakule. See on põhjus, miks on vaja teha täiendavaid uuringuid selles valdkonnas. Eesmärgiks on aidata lõppkasutajatel kui tarkvaraarendajal luua tarkvara, tagades usaldusväärseid turvalisuse ja konfidentsiaalsuse tagatise [12].

## **4.2 Lõppkasutajate programmeerimine**

Lõppkasutajate programmeerimine ehk end-user programming (EUP) on programmeerimine, mis on pigem mõeldud tulemuste saamiseks, mitte tarkvara arendamiseks. Kui professionaalse programmeerimise eesmärgiks on mingi tarkvara loomine, et seda kasutaksid teised kasutajad, siis EUP metoodika kasutaja eesmärgiks on mingi programmi või tarkvaratoode praktiline kasutamine. Lõppkasutajate programmeerimine võimaldab luua näiteks mingite rakenduste laiendid või iseseisvad uued programmid, mis töötavad tarkvarast eraldi [12].

### **4.2.1 Visuaalne programmeerimine**

Rääkides lõppkasutaja programmeerimisest on eriti tähtis mainida visuaalset programmeerimist. Visuaalne programmeerimine on selline tarkvara arendusviis, mis võimaldab graafilist kasutajaliidest kasutades luua tarkvararakendust. Graafilises kasutajaliideses võivad olla käsud, elemendid, vidinad, šabloonid jne. Selline programmeerimisviis on kõige arusaadavam mitte IT-inimestele ning annab võimaluse mitteprofessionaalist tarvaraarendajale olla integreeritud mingi IT-firma arendusprotsessi.

## **4.3 Meta-design**

Meta-disain iseloomustab eesmäärke, võtteid ja protsesse, mille abil luua uus meedia ja keskkond, mis võimaldab “probleemide omanikele” (või lõppkasutajatele) tegutseda disaineritena. Põhiliseks meta-disaini eesmärgiks on luua sotsiaal-tehniline keskkond, mis võimaldaks kasutajatel aktiivselt osaleda süsteemide pidevas arendamises [24].

Meta-disain laiendab meie traditsioonilist süsteemiarenduse mõistet, kuna selline metoodika lülitab kasutajaid arendusprotsessi nagu kaasarendajad (co-designers). Nad tegutsevad mitte ainult disaini iteratsioonis vaid kogu protsessi jooksul. Meta-disain pakub kasutajatele võimalusi, tööriistu ja sotsiaalseid struktuure, et laiendada süsteemi nende vajaduste rahuldamiseks [24].

## 5 Low-code/no-code

Antud lõputöö osa eesmärgiks on uurida mis on low-code/no-code platvormid, millised low-code/no-code platvormid on olemas, uurida nende platvormide tööriistu ja funktsioone. Antud peatükki kirjutatakse low-code/no-code platvormide võimalustest ning nende populaarsusest tänapäeval.

### 5.1 Low-code/no-code platvormid

Varem oli piisav, kui arendajal oli programmeerimiskogusi ja teadmisi uuetest tööriistadest. Tänapäeval peab kõigil olema kokkuvõttes kõik, alates suhtlemisostkustest kuni äritegevuse kogemuseni. Kuidas jõuda nii, et arendada kõiki vajalikke oskusi minimaalse aja jooksul ning väikese summa eest. Forrester Research'i peamine analüütik Jeffrey Hammond osutas, et ta näeb üha rohkem arendajaid konverentsidel ilma traditsioonilise tarkvaraarendamise taustadeta. Põhjuseks on vahendid, mis muudavad koodi kirjutamist lihtsamaks kui kunagi varem – low-code/no-code platvormid [16].

Low-code/no-code platvormide populaarsus on tingitud sellest, et kõigepealt low-code/no-code platvormid lihtsustavad tarkvaraarendaja arendamisprotsessi. Rääkides programmeerimisest on väga tähtis see, et programmeerimiskursused maksavad üsna palju. Et saada kõrgel tasemel programmeerijaks pole piisav teada ainult ühte programmeerimiskeelt, mis omakorda kohustab tarkvaraarendajat maksma mitme kursuste eest üsna kopsakat summat. Kodeerimisele keskendumise puhul rõhutatakse nii probleemi lahendamiseks vajaliku "õige" meetodi valimise tähtsust kui ka probleemi mõistmise tähtsust. Low-code/no-code platvormid võimaldavad tarkvaraarendajal pöörata rohkem tähelepanu probleemi mõistmisele, mis omakorda annab parema tulemust.

Low-code ja no-code erinevus ei seisne selles, kas inimesed tahavad kodeerida või mitte. Erinevus on pigem selles kes ja milleks kasutavad neid platvorme. No-code'i "nurgas" on "citizen developers" ehk kodanikud – ärikliendid, kes saavad luua funktsionaalseid, kuid üldiselt piiratud rakendusi ilma koodi kirjutamiseta. Low-code'i "nurgas" on professionaalsed arendajad, kes low-code platvormide abil lihtsustavad oma tööd kas vähese või ilma käsitsi kodeerimiseta [19].

Low-code/no-code platvormid võimaldavad luua rakendusi kasutades graafilisi kasutajaliideseid ja konfiguratsioone traditsioonilise programmeerimise asemel.



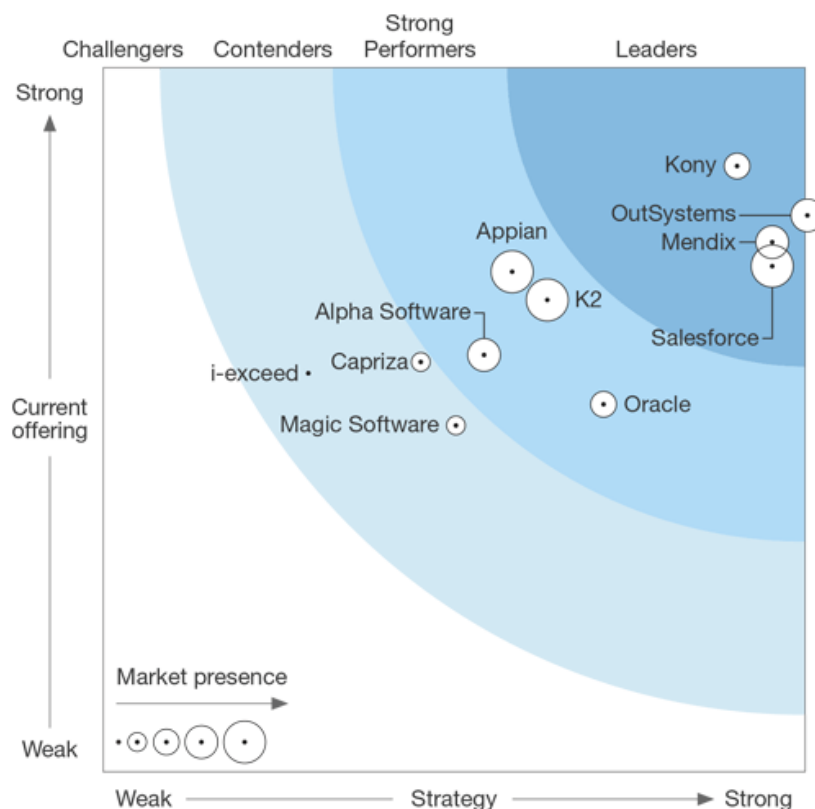
Platvormid võivad keskenduda andmebaaside, äriprotsesside või kasutajaliideste kavandamisele ja arendusele. Üldjuhul low-code/no-code platvormid peavad andma kasutajatele lihtsa kasutajaliidese, mille abil nad saavad samm-sammult luua rakendusi. Samuti peavad platvormide tööriistad lihtsustama IT arendusprotsessi, kuid samal ajal andma vajalikku funktsionaalsust.

Ettevõtted loovad liiga palju rakendusi töötajatele. 451 Research ettevõtte uuring näitas, et peaaegu pooled suurtest ettevõtetest töötavad välja 10-29 mobiilirakendust aastas oma töötajatele. Vastavalt Harmon.ie aruandele peab keskmine töötaja igal tööpäeval tööl kasutama 9,4 uut rakendust ja 74% -l on korruga avatud viis rakendust [17]. Kuna ettevõtted sageli kasutavad programmeerijate teenuseid tööprotsessi jaoks vajalike rakenduste arendamiseks, oleks mõistlikum, kui töötajad saaksid ise vajalikke tarkvaraarendusi välja töötada. Võimalik, et töötaja teeks asja lihtsamalt ja paremini, kui professionaalsete tarkvaraarendajate teenuste kasutades ning samal ajal palju raha selle eest kulutades. Just selleks võiksid low-code/no-code platvormid olla kasulikud. Need platvormid võimaldavad professionaalsetel tarkvaraarendajatel lihtsustada ja kiirendada tarkvara realiseerimise protsessi, kuid annavad ka võimaluse ärikasutajatele, kes kas on kunagi koodi kirjutamisega kokku puutunud või mitte, luua endale või teistele kasutajatele tarkvaraarendust või osaleda mingi firma tarkvara arendusprotsessis.

## **5.2 Low-code mobiilarenduse realiseerimiseks**

Seoses sellega, et antud töö praktilises osas realiseeritakse mobiilirakendust, edaspidi uuritakse low-code/no-code platvorme, mis kõige paremini sobivad mobiilarendamise jaoks.

The Forrest Wave - üks mõjukamaid uurimis- ja nõustamisfirmasid maailmas avaldas oma uuringu, mis näitab 2017. aasta esimese veerandi kõige paremad low-code platvormid mobiilarendamiseks.

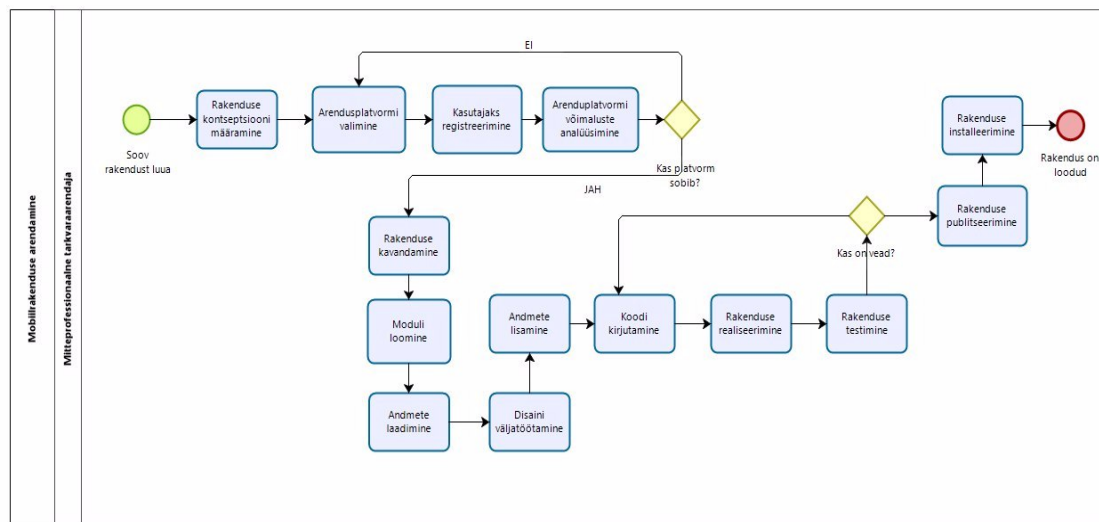


Joonis 8. Low-code platvormid mobiilarenduseks 2017. aasta alguses [20].

Turul kohaloleku ja tugevuse järgi kõige paremad platvormid low-code mobiilarendamiseks 2017. aasta alguses olid OutSystems, Mendix ja Salesforce.

Arvutiajakiri pcmag.com uuris populaarsemate low-code platvormide funktsionaalsust. Vastavalt pcmag.com uuringule OutSystems, Mendix ja Salesforce platvormid pakuvad: tasuta prooviversioon, samm-sammult juhised, Drag-and-Drop kasutajaliides, vormi ehitaja, eelinstallitud mallid, protsessi/töövoogude modelleerimine, hõlpsalt kohaldatav rakenduse kasutajaliides, andmebaasi integreerimine, kolmanda osapoole rakenduste realiseerimine, rakenduste turg jne.

Edasi on esitatud Bizagi diagramm, mis on koostatud antud töö autori poolt. Diagrammis on esitatud mobiilirakenduse realiseerimise protsess low-code platvormi abil.



Powered by  
bizagi  
Modeler

Joonis 9. Mobiilirakenduse realiseerimine low-code platvormi abil.

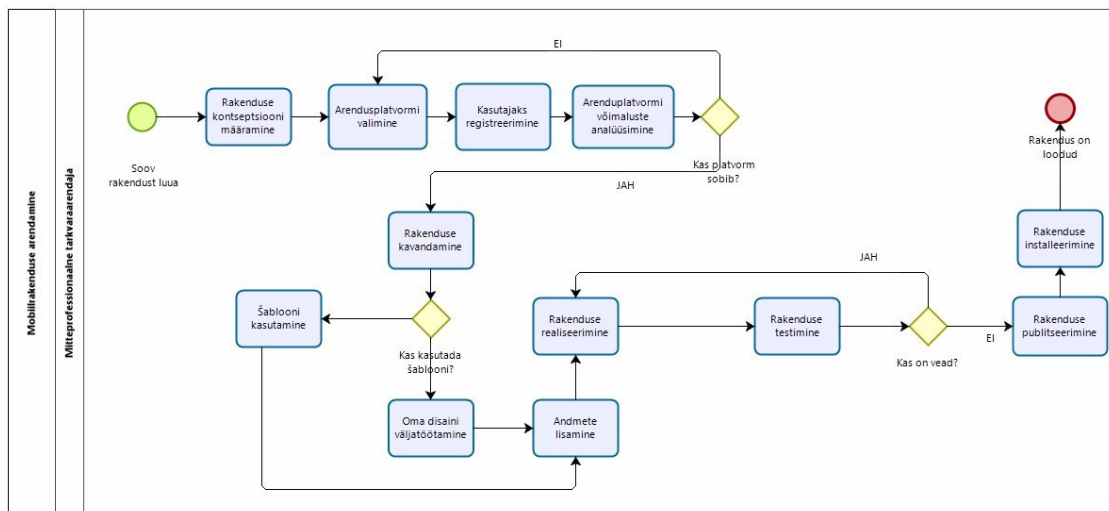
Mobiilirakenduse arendajaks on mitteprofessionaalne arendaja, kes kasutades mingi low-code platvormi tööriistu loob mobiilirakendust. Diagrammis on näha, et platvormi kasutaja realiseerib nii fron-end'i kui ka natuke back-end'i ehk puutub koodi kirjutamisega kokku.

Sellest järeldub, et low-code ei ole täiesti mitte IT-valdkonnast inimestele sobilik.

### 5.3 No-code mobiilarenduse realiseerimiseks

Kui low-code platvormid on rohkem mõeldud professionaalsete koodi tasemel tarkvaraarendajatele ning nende platvormide põhiülesanne on muuta tarkvaraarendajate tööd lihtsamaks ja kiiremaks, siis no-code platvormidest rääkides võib mainida ka tavalisi kodanike, kes ei ole koodi kirjutamisega kunagi kokku puutunud. No-code arendamine on pigem mõeldud iseseisvaks arendamiseks ärikasutajatele, kellest saab niinimetatud “citizen developer”. No-code platvormid pärinevad low-code platvormidest, kuid no-code on pigem visuaalne arendamine. Visuaalse arendamise all mõistetakse mingit platvormi, kus on juba valmis (platvormi arendaja poolt pakutud) komponendid, mis kasutaja paigutab tarkvara arendamise tööalasse. Komponentideks on tekstiväljad, nupud, pildi lisamise komponendid, linkide lisamise komponendid jne. Nendest komponentidest koostab ärikasutaja tarkvararakenduse.

Järgmisel autori poolt tehtud Bizagi diagrammil on esitatud tarkvara realiseerimine no-code platvormi abil.



Powered by  
bizagi  
Modeler

Joonis 10. Mobiilirakenduse arendamine no-code platvormi abil.

Eeldatakse, et tarkvara loob tavaline ärikasutaja, kes pole kunagi programmeerimisega kokku puutunud. Kogu arendusprotsessis on arendaja vaba oma valikutest. Samuti eeldab no-code platvormide abil tarkvararakenduse loomine ärikasutajale võimalust kasutada platvormi paigutatud šabloone, mis omakorda säästab aega tarkvara realiseerimiseks. Üldiselt, arendaja tegeleb ainult front-end'i väljatöötamisega.

### 5.3.1 No-code platvormide võrdlus

Hetkel turul on üsna palju võimalusi no-code arendamiseks. Järgnevalt on koostatud tabel, kus võrreldakse erinevaid no-code platvorme, nende võimalusi ning hindu.

Tabel 1. No-code platvormide omaduste võrdlemine [21].

Platvorm	Omadused ja võimalused	Operatsioonsüsteemid ja telefonid	Plaanid ja hinnad
shoutem	Ei vaja koodi kirjutamist Võimalus valida rakenduse kategooriat Valmis kasutajaliidese komponendid CMS Andmebaasi olemus Analüüsimise võimalus	Android, iPhone, HTML5	Tasuta proov Basic – \$19.90/kuus Advanced - \$49.90/kuus Unlimited - \$119.90/kuus

Appy Pie	Ei vaja koodi kirjutamist Võimalus valida rakenduse kategooriat Google play ja iTunes Valmis kasutajaliidese komponendid	Windows 8, Android, iPhone, teised nutitelefonid	Tasuta BASIC – \$7/kuus GOLD - \$19/kuus PLATINUM - \$33/kuus
mobincube	Ei vaja koodi kirjutamist Võimalus valida rakenduse kategooriat Visuaalne kasutajaliides Online shop PayPal Payments Google maps Calendars Phone Calls Navigation bars Push notifications Andmebaas	Windows, Android, iPhone, HTML5	Starter - \$0/kuus Home - \$2.99/kuus Standart - \$9.99/kuus Freelancer - \$49-99/kuus Agency - \$99.99/kuus
GoodBarber	Ei vaja koodi kirjutamist Google Market ja App Store	Android, iPhone	FREE Trial Standart - \$16/kuus Full - \$32/kuus Advanced - \$48/kuus
Infinite monkeys	Ei vaja koodi kirjutamist Google Market ja App Store Visuaalne kasutajaliides	Android, iPhone, HTML5	Free Bananas – FREE Go Ad Free - \$1/kuus Best Value - \$9/kuus

Oli valitud 5 platvormi, mis võimaldavad luua rakendusi nii Android kui ka iOS operatsioonsüsteemidele. Ükski platvormidest ei vaja koodi kirjutamist, seega nad võimaldavat mitteprofessionaalset tarkvaraarendajale luua rakendust, kasutades platvormi poolt pakutud visuaalset kasutajaliidet. Antud no-code platvormid võimaldavad kiiresti publitseerida rakendust Google Play ja App Store rakenduste poodidesse. Arendusplatvormid pakuvad nii tasuta kui ka tasulisi versioone, mis sisaldavad laiemat funktsionaalsust.

Mobiilirakenduse realiseerimiseks on valitud Mobincube platvorm, kuna platvorm vastab kõikidele nõudmistele ning annab vajaliku funktsionaalsust mobiilirakenduse realiseerimiseks. Platvormis on tegu peamiselt front-end'i väljatöötamisega. Kasutaja ei pea koodi kirjutama ning arendab oma tarkvara visuaalse kasutajaliidese kaudu. Üldiselt platvormi kasutaja arendab ainult front-end'i. Mobincube platvormi back-end koosneb peamiselt PHP keelest. Samuti peab Mobincube platvormi back-end'i arendaja oskama testimist, objektorienteeritud programmeerimist, MVC raamistiku kasutamist ning üldse omama kogemust koodi kirjutamises. Hinnatakse ka teadmisi programmides Git, MySQL, Docker, Vargant, Capistrano, Node.js, Redis [23].

## **6 Mobiilrakenduse realiseerimine**

Antud lõputöö osa kirjeldab algklasside jaoks mobiilrakenduse realiseerimist no-code platvormi abil. Algklassi õpetaja eesmärgiks on no-code platvormi tööriistu kasutades luua algklassi õpilastele mobiilrakendus spetsialisti abiga. Rakenduse sisuks on algkooli (1-4. klass) kõik reeglid ainetel matemaatika, eesti keel, vene keel ja muusika. Põhilise osa tööst rakenduse realiseerimisel teeb mitteprofessionaalne tarkvaraarendaja ehk õpetaja, kes ei ole kunagi kokku puutunud tarkvara loomisega. Lähtudes osas [3.3 Mobiilrakendus] analüüsitud operatsioonsüsteemide populaarsusest valiti Android OS rakenduse publitseerimiseks.

### **6.1 Sobilik tarkvara**

Eelnevalt oli testitud 5 no-code platvormi ning kõikidest platvormidest sai valitud kõige sobivam mobiilrakenduse realiseerimiseks – Mobincube.

#### **6.1.1 Mobincube platvormi kasutajamugavus**

Mobincube platvorm on üsna lihtsalt arusaadav programmeerimisvõõra inimese jaoks. Antud arendusplatvorm ei nõua koodi kirjutamist. Mobincube platvormis on mugav visuaalne kasutajaliides, kus on pakutud kõik vajalikud komponendid lihtsama rakenduse arendamiseks. Mobincube on pakutud mitmetes keeltes, mis võimaldab igal kasutajal kiiresti ja mugavalt seda platvormi omandada. Kui teised no-code platvormid pakuvad ainult ajutise tasuta prooviversiooni, siis Mobincube'is saab piiramatult kasutada tasuta versiooni. Tarkvara arendamiseks sai valitud tasuta versioon, kuna seal pakutakse vajalikku funktsionaalsust selle rakenduse realiseerimiseks. Platvorm võimaldab luua mobiilrakendust erinevate operatsioonsüsteemidega nutitelefonidele, sealhulgas ka Android operatsioonsüsteemile, mis valiti rakenduse publitseerimiseks.

#### **6.1.2 Mobincube platvormi võimalused**

Mobincube platvormis iga kasutaja saab luua endale sobiva rakenduse: kas hariduslik, ärirakendus, mängu või meelelahutusega seotud rakendus, rakendus kodu, restorani või tervise jaoks jne. Tarkvaraarendaja saab vabalt valida kas teha rakendust nullist oma disainiga või kasutada juba pakutud šabloone. Mobincube platvormis on võimalik luua online-poodi, suhelda kasutajatega, lisada kaardid, paigutada videot või muusikat,

integreerida kolmanda osapoole rakendust, luua suuri andmebaase, saata SMS sõnumeid, teha telefonikõnesid ja palju teisi võimalusi pakub antud platvorm.

## **6.2 Mobiilirakenduse kavandamine**

Eesmärk: luua no-code Mobincube platvormi abil rakendust algklasside jaoks, kus on kogutud kõik Kannuka Kool algkoolis kasutavad reeglid ainetel vene keel, matemaatika, eesti keel ja muusika.

Sihtgrupp: Kannuka Kool algkooli õpilased

Metoodika:

- Mobincube platvormi võimaluste uurimine
- Materjali kogumine ja jagamine teemadeks
- Õpetaja koolitamine ja juhendamine kogu tööprotsessi käigus
- Mobiilirakenduse realiseerimine
- Mobiilirakenduse testimine ja publitseerimine
- Mobiilirakenduse proovimine/kasutamine
- Analüüs ja hinnangu andmine

Tähtajad: rakenduse realiseerimine 20 tööpäeva

Ressursid: Mobincube platvorm, õpikud, e-materjalid

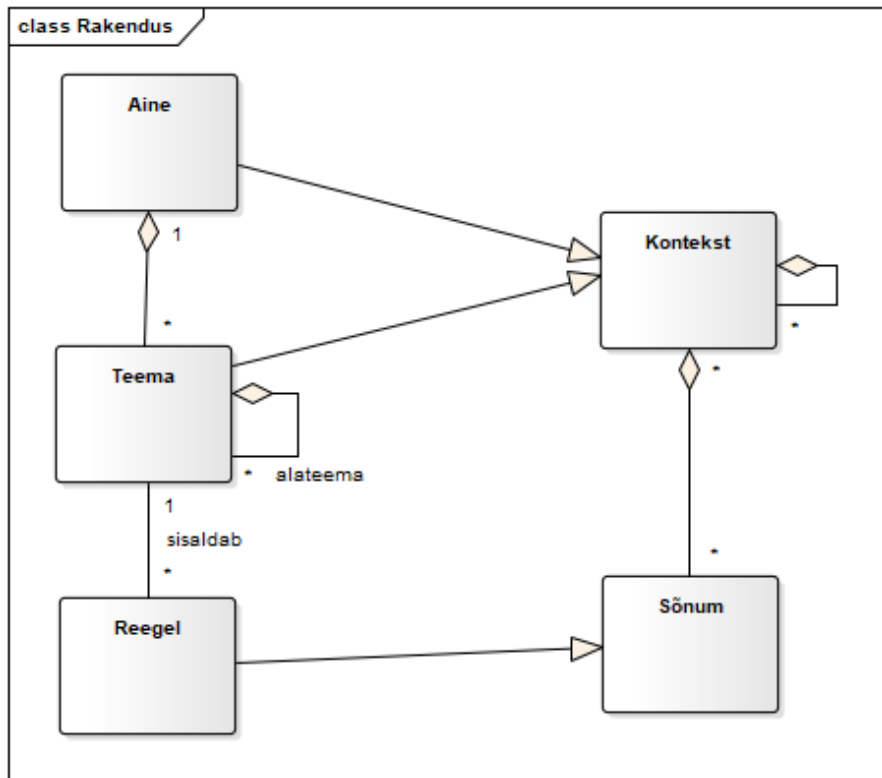
Operatsioonsüsteem: Android OS

Testimine: Mobincube platvormi testimine

Analüüs: mobiilirakenduse kasutamine tundides, küsitlus õpilaste ja õpetajate jaoks



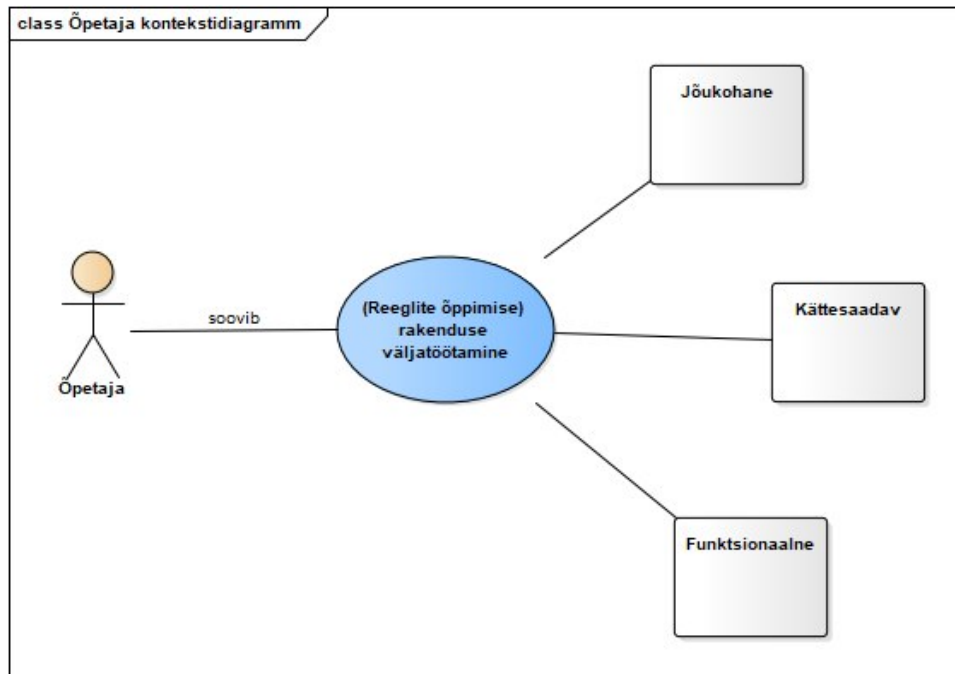
Edasi on esitatud rakenduse kontseptsiooni domeenimudel.



Joonis 11. Domeenimudel.

Domeenimudel kirjeldab rakenduse kontseptsiooni. Rakendus on mõeldud reeglite õppimiseks, seega sisaldab ained. Igas aines võib olla mitu teemat ning teema võib sisaldada alateemasid. Teema või alateema sisuks on reeglid, mis omakorda on kindlat tüüpi sõnumid. Igal sõnumilb on oma kontekst. Aine ja teema on kindlat tüüpi kontekstid. Sama sõnum võib (üldjuhul) sobida rohkem kui ühte konteksti.

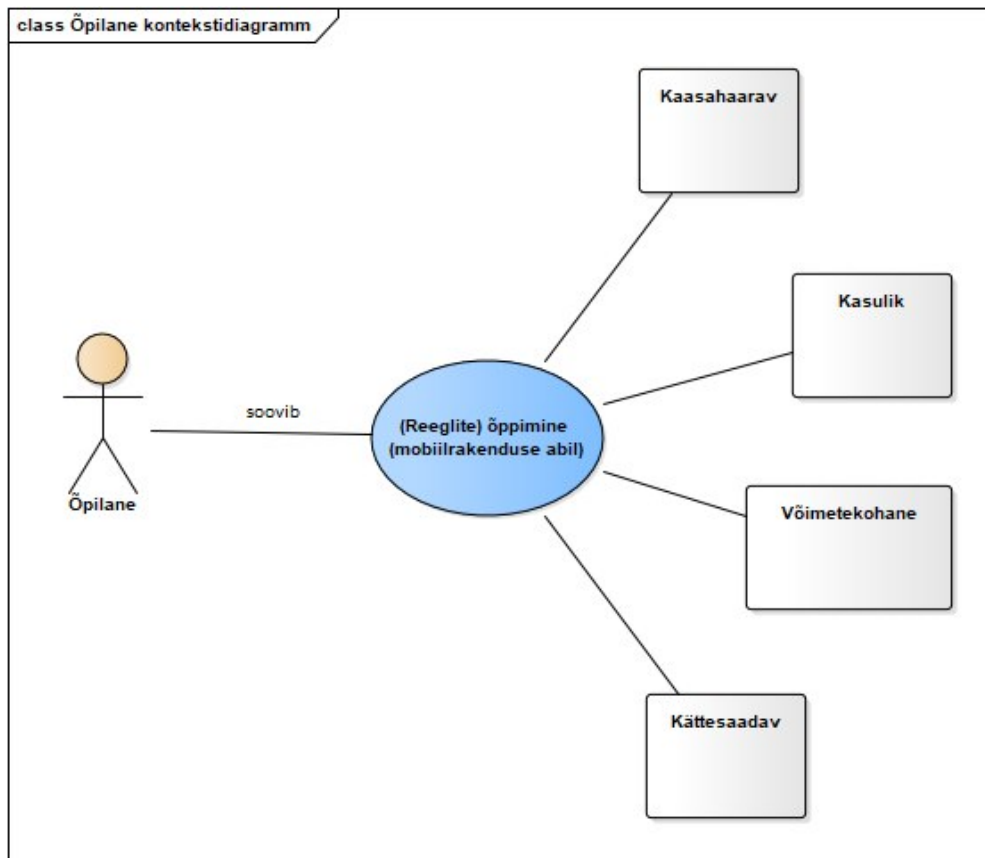
Järgnevalt on toodud ärikasutusjuhtude kontekstidiagramm, mis sisaldab rakenduse väljatöötamise vaadeldavat tegevust ühe ärikasutusjuhuna (mis väljendab õpetaja eesmärki platvormi kasutamisel).



Joonis 12. Õpetaja eesmärgid platvormi kasutamisel.

Diagrammi peamiseks tegelaseks on õpetaja kes viib läbi reeglite õppimise rakenduse (või muu õpetajale jõukohase rakenduse) väljatöötamise. Selline tegevus peab olema õpetajale jõukohane, kuna õpetaja eriala üldse ei puutu tarkvara realiseerimisega kokku. Samuti mobiilirakenduse väljatöötamine peab olema kättesaadav ehk tasuta või minimaalse tasuga, kuna õpetaja teeb rakendust vabatahtlikult oma klassi jaoks ning keegi seda ei sponsoreeri. Mobiilirakenduse realiseerimine peab olema funktsionaalne, mis tähendab, et väljatöötamiseks peavad olema kõik funktsionaalsused, mis võiksid olla vajalikud rakenduse loomiseks.

Järgnevalt on toodud ärikasutusjuhtude kontekstidiagramm, mis näitab (reeglite) õppimise tegevust mobiilirakenduse abil ühe ärikasutusjuhuna.



Joonis 13. Õpilase eesmärgid rakenduse kasutamisel.

Diagrammil on esitatud üks tegelane – õpilane. Tema eesmärgiks on reeglite (või muud tüüpi sõnumitena esitatud materjali) õppimine õpetaja poolt väljatöötanud mobiilirakenduse abil. Õpilase jaoks mobiilirakenduse kasutamine peab olema kaasahaarav, et tema saaks seda rakendust huviga kasutada nii koolis kui ka kodus. Huvitav reeglite õppimine võib mõjutada õpilaste õppeedukust. Rakendus peab olema kasulik õpilasele ning andma talle uue teadmisi, mida ta saaks kergesti igal hetkel omandada. Samuti reeglite õppimine peab olema õpilasele võimetekohane. See tähendab, et rakenduses asuvad ainult need reeglid, mis on mõeldud algklassi õppekava täitmiseks ning need on selgelt sõnastatud, et õpilane kergesti saaks reeglite sõnumitest aru. Rakendus peab olema õpilasele kättesaadav (eeldatavasti tasuta).

## 6.3 Realiseerimise etapid

Rakenduse realiseerimise etapid:

### 1. Rakenduse kavandamine

Õpetaja määrab mobiilirakenduse kontseptsiooni ja kavandab tuleviku rakendust.

Aeg: 1 tööpäev

### 2. Arendusplatvormi valimine

Spetsialist valib arendusplatvormi, arvestades õpetaja poolt määratud rakenduse kontseptsiooni ning tema soovitusi.

Aeg: 2 tööpäeva

### 3. Platvormi võimaluste analüüsimine

Selles etapis uurib õpetaja koos lõputöö autoriga platvormi, mis oli valitud mobiilirakenduse realiseerimiseks. Selgitatakse välja milliseid võimalusi pakub antud platvorm ning milliseid tööriistu on olemas.

Aeg: 2 tööpäeva

### 4. Materjali (reeglite) kogumine

Erinevate ainete õpetajad pakuvad informatsiooni (reeglid), mis edaspidi tuleb rakenduses kasutada.

Aeg: 3 tööpäeva

### 5. Materjali teemadeks jagamine

Kogutud informatsioon õpetaja jagub teemadeks ning vajadusel alateemadeks.

Aeg: 1 tööpäev

### 6. Rakenduse reeglitega täitmine

Õpetaja spetsialisti juhendamise all täitab rakendust reeglitega.

Aeg: 7 tööpäeva

### 7. Disaini väljatöötamine

Õpetaja töötab välja kõige sobilikuma disaini mobiilirakenduse jaoks.

Aeg: 2 tööpäeva

### 8. Rakenduse testimine

Õpetaja testib rakendust Mobincube platvormi abil.

Aeg: 1 tööpäev

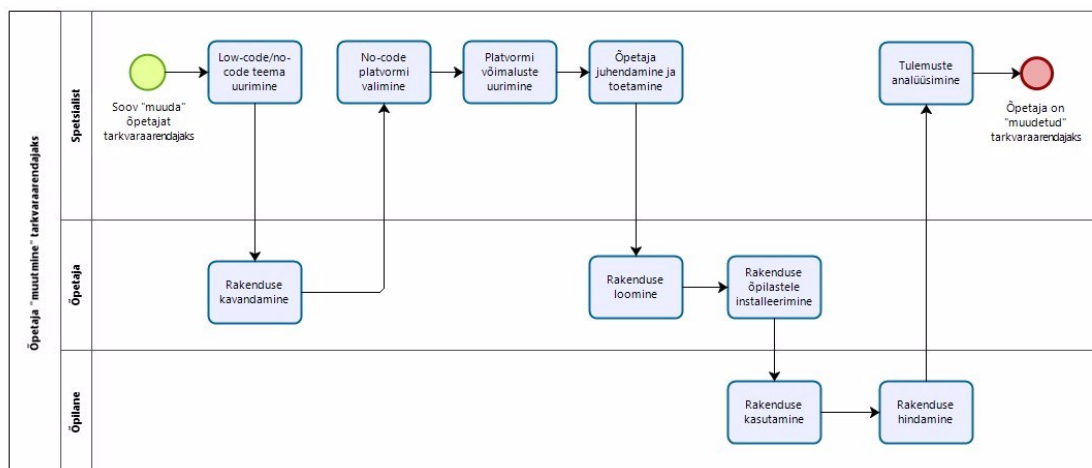
### 9. Rakenduse publitseerimine

Õpetaja spetsialisti juhendamise all publitseerib rakendust Android operatsioonisüsteemile.

Aeg: 1 tööpäev

Antud töö praktilise osa (mobiilirakenduse loomine) on esitatud autori poolt koostatud Bizagi diagrammina.

Tegevuste diagramm on järgmine:



Powered by  
bizagi  
Modeler

Joonis 14. Õpetaja “muutmine” tarkvaraarendajaks.

**Tegevus:** Low-code/no-code teema uurimine

Lõputöö autor ehk spetsialist uurib low-code/no-code teemat ning uurimise lõppus otsustab kuidas tarkvara realiseerida.

**Tegevus:** Rakenduse kavandamine

Õpetaja kavandab tuleviku mobiilirakendust ehk määrab rakenduse kontseptsiooni ja sisu.

**Tegevus:** No-code platvormi valimine

Spetsialist valib no-code platvormidest kõige sobivama mobiilirakenduse realiseerimiseks.

**Tegevus:** Platvormi võimaluste analüüsimine

Spetsialist analüüsib valitud no-code platvormi võimalusi ja tööriistu, et mitteprofessionaalne platvormi kasutaja ehk õpetaja saaks tööprotsessist aru ning edaspidi saaks iseseisvalt rakendust arendada.

**Tegevus:** Õpetaja juhendamine ja toetamine

Spetsialist juhendab ning toetab õpetajat kogu tööprotsessi käigus.

**Tegevus:** Rakenduse loomine

Õpetaja loob mobiilirakendust no-code platvormi abil. Õpetaja spetsialiste juhendamise all töötab disaini välja, täidab rakendust reeglitega ning testib loodud rakendust.

**Tegevus:** Rakenduse õpilastele installeerimine

Õpetaja installeerib loodud rakendust õpilaste nutitelefonidele.

**Tegevus:** Rakenduse kasutamine

Õpilased kasutavad rakendust koolitundides.

**Tegevus:** Rakenduse hindamine

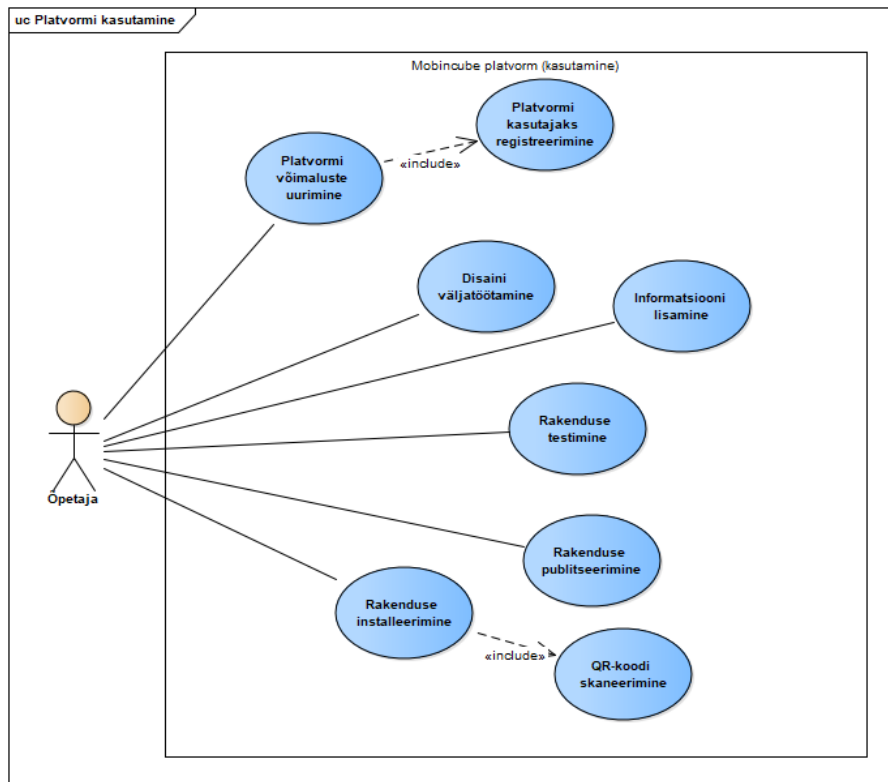
Õpilased hindavad loodud rakenduse mugavust.

**Tegevus:** Tulemuste analüüsimine.

Küsitluse vastuste ning SWOT analüüsi põhjal analüüsib spetsialist loodud mobiilirakenduse kasutamist ja mugavust.

Diagrammi lõpptegevuseks on «Õpetaja on «muudetud» tarkvaraarendajaks». See tähendab, et õpetaja saab nüüd luua sarnast tüüpi mobiilirakendusi ehk informatiivseid rakendusi üsna piiratud funktsionaalsusega. Uue kontseptsiooni ja funktsionaalsusega rakenduse loomise puhul õpetaja peab olema õpetatud ja uuesti juhendatud. Samuti õpetaja töötab ainult Mobincube platvormis, seega uue platvormil, kuid sama tüüpi rakenduse realiseerimise puhul peab mitteprofessionaalset tarkvaraarendaja olema juhendatud.

Edasi on toodud kasutusjuhtude diagramm õpetaja kui (Mobincube) platvormi kasutaja põhilised tegevuseesmärgid. Need eesmärgid on esitatud kasutusjuhtudena.



Joonis 15. Õpetaja kasutajaeesmärgid Mobincube platvormis.

**Kasutusjuht:** Platvormi võimaluste analüüsimine

Õpetaja analüüsib, kas antud platvormi tööriistad ja võimalused sobivad mobiilirakenduse realiseerimiseks või mitte ning proovib neid.

**Kasutusjuht:** Platvormi kasutajaks registreerimine

Õpetaja loob Mobincube platvormis endale kontod, et edaspidi arendada oma kontos rakendusi.

**Kasutusjuht:** Disaini väljatöötamine

Õpetaja muudab olemasoleva või loob uue disaini oma rakenduse jaoks.

**Kasutusjuht:** Informatsiooni lisamine

Õpetaja täidab rakendust vajalikku informatsiooniga.

**Kasutusjuht:** Rakenduse testimine

Õpetaja testib antud rakendust Mobincube platvormi testimise funktsiooni kaudu.

**Kasutusjuht:** Rakenduse publitseerimine

Õpetaja publitseerib loodud rakendust sobivale operatsioonisüsteemile.

**Kasutusjuht:** Rakenduse installeerimine

Õpetaja installeerib ja seadistab loodud mobiilirakendust nutitelefonile.

**Kasutusjuht:** QR-koodi skaneerimine

Õpetaja skaneerib QR-koodi, et laadida rakendust.

## 6.4 Õpetamine

Platvormid võimaldavad tavakasutajatel olla integreeritud mingi firma arendusprotsessi või arendada tarkvara iseseisvalt. Integreerimine arendusprotsessi tähendab seda, et tavakasutaja teeb mingit kindlad tööülesanded, mida on võimalik teha low-code/no-code platvormide abil. Kuna tegemist on mitteprofessionaalse tarkvaraarendajaga, kes hakkab arendada rakendus iseseisvalt, siis on vaja õpetada kasutajat kuidas platvormis töötada. Õpetamisele kulutatakse üsna palju aega, kuna inimene, kes pole kunagi IT-sfääriga kokku puutunud peab üsna piiratud aja jooksul saama tarkvaraarendajaks. Esialgu räägitakse üldiselt mobiilirakenduse realiseerimisest, publitseerimisest ning installeerimisest. Edasi hakatakse uurima platvormi võimalusi ning tööriistu.

## 6.5 Integreerimine

Tänapäeval üha rohkem arutatakse rakenduste integreerimise küsimust. Tavaliselt vajab ettevõtte rohkem kui ühte infosüsteemi/rakendust, et rahuldada kõiki nõudeid. Keskmise suurusega ja suured ettevõtted kasutavad oma tegevuses kümneid või isegi tuhandeid erinevaid rakendusi. Nende rakenduste integreerimine aitab suurendada tüüpiliste ülesannete lahendamise kiirust ning tagada ülesannete tähtajaliselt täitmist. Samuti rakenduste integreerimine parandab ülesannete lahendamise kvaliteeti.

Low-code platvormide puhul on võimalik integreerida rakendustesse teiste arendajate kirjutatud koodi. Selleks on näiteks järgmised rakendustarkvara liidesed: Software as a Service, Web Service, Application Programming Interface (API), Containers and MicroServices ja paljud teised. Need sisaldavad tööriistade komplekti, protokolle, valmis klasside kogumit, struktuure, konstante ehk kõike, mis on vaja rakendustarkvara ehitamiseks.

No-code platvormide puhul sellist võimalust hetkel ei ole, kuid saaks olla. Selle võimaluste puhul saaksid mitteprofessionaalidest tarkvaraarendajad luua täielikult funktsionaalseid rakendusi.

Rakenduste koostöö jaoks kasutatakse tavaliselt selliseid meetodeid nagu failide vahetus, ühine andmebaas ja asünkroonne sõnumivahetus. Ühise andmebaasi puhul mitmel



süsteemil on üks andmebaas. Antud rakendust “MyStudy” saab integreeriga läbi SQL andmebaasiga ühendamise (Table tüüpi elementide sidumine SQL andmebaasiga).

Kuna tegemist on ainult reeglite õppimisega, on võimalik luua ka teisi rakendusi õppimisprotsessi jaoks, kuid mitte kõik pole võimalik realiseerida antud platvormi abil. Mobiilirakendust MyStudy võib laiendada, lisades reeglite sisestamise võimaluse õpetajale, mis on võimalik arendada low-code platvormide abil. Peale reeglite koostamisest ja õppimisest on võimalik luua rakendus, mis võimaldaks õpetajale koostada reeglite õppimiseks teste/kontrolltöid ning seal ka neid lahendada. Sellist tüüpi rakendust no-code platvormid ei toeta. Kui sellised rakendused on valmis, siis oleks võimalik integreerida MyStudy rakendust. Üldiselt kõik algkooli õppimisprotsessiga seotud rakendused võiksid kasutada MyStudy mobiilirakendust, kuna see on universaalne rakendus, mis toetab ainult informatsiooni edastamist ning võimalust seda informatsiooni õppida ja igal hetkel kasutada.

## 7 Analüüs

Antud töö järgmises osas kirjutab autor loodud mobiilirakendusest ning analüüsib Kannuka Kooli algklasside õpilaste ja õpetajate tagasisidet ja hinnanguid mobiilirakendusele. Samuti arutleb autor antud osas mobiilirakenduse edasise arengu üle. Loodud rakenduse osasid ekraanipilte saab vaadata Lisas 4.

### 7.1 Loodud rakenduse analüüs

Loodud mobiilirakendus MyStudy oli realiseeritud Mobincube no-code platvormi abil. Kuna rakenduse arendajaks on mitteprofessionaalne tarkvaraarendaja ehk algkooli õpetaja, siis kasutati Mobincube platvormi tasuta versioon, mis pakub üsna piiratud funktsionaalsust. Rakenduses on neli ainet, millised oli omakorda jagatud teemadeks. Rakenduses on lisatud pildid ja tabelid. Mobiilirakenduse realiseerimine oli jagatud etappideks ning iga etapi jaoks oli määratud aeg. Järgnevalt on toodud tabel, mis näitab rakenduse realiseerimise planeeritud ja tegelikke (täht)ageid tööpäevades.

Tabel 2. Mobiilirakenduse loomise tähtaegade võrdlemine.

Etapp	Planeeritud (tööpäev)	Tegelik (tööpäev)
Rakenduse kavandamine	1 tööpäev	1 tööpäev
Arendusplatvormi valimine	2 tööpäeva	4 tööpäeva
Platvormi võimaluste analüüsimine	2 tööpäeva	3 tööpäeva
Materjali (reeglite) kogumine	3 tööpäeva	5 tööpäeva
Materjali teemadeks jagamine	1 tööpäev	1 tööpäev
Rakenduse reeglitega täitmine	7 tööpäeva	10 tööpäeva
Disaini väljatöötamine	2 tööpäeva	2 tööpäeva
Rakenduse testimine	1 tööpäev	1 tööpäev
Rakenduse publitseerimine	1 tööpäev	1 tööpäev
<b>Kokku:</b>	20 tööpäeva	28 tööpäeva

Arendusplatvormi valimine võttis rohkem aega, kui oli planeeritud, kuna oli testitud mitu platvormi. Oli proovitud erinevate platvormide tööriistu ja võimalusi: platvormi mallid,

arendamine nullist, mobiilirakenduse publitseerimine jne. Seega platvormi võimaluste analüüsimine kestis ka kauem. Informatsiooni kogumisele sai kulutatud rohkem aega, kui oli planeeritud, kuna rakenduse kavandamise etapis oli planeeritud lisada ainult matemaatika ja vene keele reegleid, kuid töö käigus sai lisatud teised ained nagu muusika ja eesti keel. Kõige suurim vahe on rakenduse reeglitega täitmise etapis, mis on päris ilmne, kuna tarkvara arendusprotsessi käigus tekkisid erineva iseloomuga probleemid, takistused, ettepanekud ning uued ideed. Üldiselt arendusprotsess kestis kaheksa päeva kauem, kui oli planeeritud. Sellised hälbed olid üsna oodatavad arendusplaani alguses, kuna tegemist on mitteprofessionaalset tarkvaraarendajaga ning täiesti uue arendusplatvormiga. Kuna rakenduse arendajaks oli tavaline kasutaja ning rakenduse loomine oli vabatahtlik, siis rangeid ajalisi piiranguid ei olnud ning kõrvalekalded graafikust olid võimalikud.

## 7.2 SWOT analüüs

Järgmiselt on toodud SWOT analüüs mobiilirakenduse kasutamise kohta, mis näitab millised on tugevused ja nõrkused mobiilirakenduse kasutamisel, milliseid võimalusi annab rakenduse kasutamine ning millised ohud võivad tekkida selle rakenduse kasutusele võtmisega tundides.

Tabel 3. Mobiilirakenduse kasutamise SWOT analüüs.

Tugevused	Nõrkused
<ol style="list-style-type: none"> <li>1. Esimene mobiilirakendus algkooli õpilastele Sillamäe linnas</li> <li>2. Infotehnoloogiate kasutamine algkoolis</li> <li>3. 4 erinevat ainet</li> <li>4. Kõik algkoolis käsitletud reeglid</li> <li>5. Kõik vajalikud reeglid ühes kohas</li> <li>6. Kiire ja mugav</li> <li>7. Õpetajad säästavad aega ja raha printimisele</li> <li>8. Tasuta laadimine</li> <li>9. Loodusesõbralik (paberist loobumine)</li> </ol>	<ol style="list-style-type: none"> <li>1. Pole otsingu võimalust</li> <li>2. Erinevad nutitelefonide gabariidid (väike kiri, pildid)</li> </ol>

Võimalused	Ohud
<ol style="list-style-type: none"> <li>1. Võimalus installeerida oma nutitelefonile</li> <li>2. Võimalus kasutada kooli väljaspool</li> <li>3. Võimalus kasutada Interneti ühenduseta</li> </ol>	<ol style="list-style-type: none"> <li>1. Nutitelefonide puudus</li> <li>2. Uue mobiilirakenduse kasutamise arusaamatus</li> <li>3. Vanemad/õpilane/õpetaja võib eelistada tavalist reeglite saamise viisi (printimine)</li> <li>4. Mingi aine või reegli puudumine rakenduses</li> <li>5. Tegevusalasse lisandub mitu uut sarnast teenust pakkuvad firma/isikut</li> <li>6. Telefonisõltuvus</li> </ol>

SWOT analüüsi raames oli määratletud mobiilirakenduse kasutamise tugevad küljed, nõrgad küljed, võimalused ja ohud. Analüüs näitas, et tugevuste osakaal on kõige suurem mis tähendab, et mobiilirakenduse MyStudy kasutamine on pigem kasulik kui kahjulik õppimisprotsessi jaoks. Need nõrkused ja ohud, mis praegu võivad tekkida mobiilirakenduse kasutamises ei takista rakenduse peamise eesmärgi täitmist – reeglite õppimist.

### 7.3 Rakenduse hindamine (küsitlus)

Rakenduse hindamiseks oli loodud küsitlus algklasside õpilaste ja õpetajate jaoks (Lisa 1). Küsitluses osales 30 algklasside õpilast ning 15 algkooli õpetajat. Küsitlus oli kirjalik ning oli koostatud vene keeles (Lisa 1 leheküljel on küsitluse küsimused eesti keeles). Küsitlusega lehed olid jagatud pärast rakenduse installeerimisest ja kasutamisest. Õpilaste küsitluse vastuse põhjal sai loodud järgmine tabel, kus on esitatud Jah/Ei küsimused ja vastuste arv.

Tabel 4. Õpilaste vastused küsitlusele mobiilirakenduse kasutamise kohta.

Küsimus	Jah	Ei	Kokku
Kas Teile meeldis mobiilirakendus, kus on kogutud kõik reeglid algkooli õpilaste jaoks?	29	1	30

Kas tahaksite kasutada seda rakendust edasi?	28	2	30
Kas antud rakendus on mugav kasutamisel?	29	1	30

Vastustest on näha, et 29 õpilastele meeldis loodud rakendus ning 28 neist tahaksid seda rakendust edaspidi kasutada. Samuti peaaegu kõik vastajad (29 õpilast) arvavad, et mobiilirakendus MyStudy on mugav tundides kasutamiseks ja reeglite õppimiseks.

Õpetajate jaoks oli tehtud teistsugune küsitlus, mille vastused on esitatud tabelina, kus on ainult Jah/Ei küsimused. Ülejäänud küsimused võimaldaksid rohkem kui ühe vastuse valimist või oma vastuse kirjutamist, seega need on kirjeldatud tekstina.

Tabel 5. Õpetajate vastused küsitlusele mobiilirakenduse kasutamise kohta.

Küsimus	Jah	Ei	Kokku
Kas Teil kunagi tekkis vajadus niisugusest mobiilirakendusest?	12	3	15
Kas antud rakendus on mugav Teie jaoks?	15	0	15
Kas Teie tahaksite kasutada antud mobiilirakendust oma klassi jaoks?	15	0	15

Õpetajate vastuste põhjal võib öelda, et küsitletute seas oli 12 õpetajat, kes tundsid vajadust niisuguse mobiilirakenduse järele. Samuti 100% õpetajatest leidsid, et rakendus on üsna mugav kasutamiseks, seega tahaksid loodud mobiilirakendust oma klassi õppimisprotsessis edaspidi kasutada.

Millised on antud rakenduse plussid Teie jaoks?

- printimiseks ja kleepimiseks aja säästmine – **15**
- kõik reeglid algkooli klasside jaoks on ühes kohas – **13**
- võimalus kasutada rakendust koolist väljas – **14**
- oma variant: unikaalsus, paberi ja tahmakasseti säästmine

Kõik 15 õpetajat hindaksid loodud rakendust 5-skaala põhjal hinnaga 5 - väga hea.

Saadud vastuste põhjal võib teha lõppkokkuvõtte, et mobiilirakendus meeldis nii õpilastele kui ka õpetajatele ning enamik vastajatest on rakendusega rahul, mis põhjustab nende soovi kasutada loodud rakendust tulevikus.

#### **7.4 Rakenduse valideerimine**

Esitatud mobiilirakenduse valideerimiseks oli läbi viidud eraldi tunnid algkoolis. Koolitunni jooksul tutvustasid õpetaja (mitteprofessionaalne tarkvaraarendaja) ja spetsialist oma tööd esialgselt algkooli õpilastele ja õpetajatele testimiseks. Esimese katse tulemused näitasid, et õpilased saaksid kergesti leida vajalikke reegleid ning omandasid kiiresti uut mobiilirakendust. Algkoolide õpetajatel oli samuti võimalus rakendust testida. Katsetamise ajal olid kõik õpetajad tootega rahul ja kavandasid selle edasist kasutamist õppimisprotsessis. Valideerimine näitas, et loodud toode on õpetajate ja õpilaste kasulik ning hästi sobib reeglite õppimiseks tuleviku õppimisprotsessis.

Lisaks õpilastele ja algkooli õpetajatele näidati seda mobiilirakendust selle kooli peaõpetajale ja direktorile. Direktori arvamus on esitatud Lisa 3 jaotises.

#### **7.5 Edasine areng**

Mobiilirakendus MyStudy oli loodud algklasside jaoks ning kõik mobiilirakenduses asuvad reeglid on ainult algkooli õpilaste jaoks. Arengu käigus tekkisid ettepanekud täiendada rakendust reeglitega põhikooli ja gümnaasiumi jaoks.

Mobiilirakenduse sihtgrupiks oli valitud vene kooli õpilased. Tuleviku eesmärgiks võib seadistada rakenduse realiseerimine eesti koolide jaoks.

Kuna mobiilirakenduse publitseerimise operatsioonsüsteemiks oli valitud Android OS, siis võiks seda rakendust edaspidi arendada ka iOS operatsioonsüsteemile, et seda rakendust kasutaksid rohkem inimesi.

Samuti rakenduse loomisel oli kasutatud Mobincube platvormi tasuta versioon. Kui arendada antud rakendust edasi ning laiendada rahvusvahelise turule, siis on võimalus kasutada Mobincube platvormi tasulist paketti, mis omakorda pakub rohkem funktsionaalsust ja võimalusi.

Lähtudes õpetaja põhitööülesannetest, mis on esitatud Joonisel 1 on võimalik luua sama tüüpi, kuid teise sisuga mobiilirakendusi. Näiteks Mobincube no-code platvormi abil saaks

õpetaja luua rakendust individuaalse õppekavaga. Individuaalse õppekava on võimalik ka MyStudy rakendusesse integreerida.

## 7.6 Rakenduse installeerimine

Rakendust MyStudy saab installeerida Android operatsioonisüsteemiga nutitelefonidele. Laadimine on tasuta. Rakenduses on olemas reklaam, mis on paigutatud Mobincube platvormiga. Rakenduse ilma Interneti ühenduseta kasutamisel reklaam puudub. Rakenduse maht on 33,67 Mb.

Edasi on toodud rakenduse installeerimise aadress (viide peab olema sisestatud nutitelefonis brauserisse) - <http://mobincube.mobi/DZH3XJ>.

Rakenduse installeerimiseks võib samuti kasutada Mobincube platvormi poolt genereeritud QR-koodi, mis on esitatud järgmise joonisena.



Joonis 16. QR-kood mobiilirakenduse MyStudy installeerimiseks.

Skaneerimiseks on võimalik kasutada järgmisi tasuta programme Android Play Market-ist: QR Code Reader, QR code scanner, QR Scanner & Barcode Scanner 2018 ja teisi.

## 8 Kokkuvõte

Käesolevas lõputöös „Mitteprofessionaalset tarkvaraarendajale töökeskkonna loomine low-code/no-code platvormil algklasside õpetaja näitel“ otsiti lahendust IT-maailma probleemile, kuidas teha nii, et mitte IT-sfäärist inimene saaks olla integreeritud tarkvara arendusprotsessi või ise luua vajalikke tarkvaralahendusi. Praegu on professionaalsete koodi tasemel tarkvaraarendajate defitsiit, seega selline võimalus nagu low-code/no-code platvormid võiksid olla probleemi lahenduseks.

Antud bakalaureusetöö eesmärgiks oli "muuta" konkreetset kooliõpetajat mitteprofessionaalset tarkvaraarendajaks, et tema saaks low-code/no-code platvormi abil luua oma õpilastele vajalikku lihtsat tüüpi tarkvararakendust. Töö oli suuremas osas eksperimentaalne, teadmata kas õpetaja saab rakenduse arendamisega hakkama ning kas loodud rakendus tuleb mugav edaspidi tundides kasutamiseks.

Oli uuritud mitmed low-code/no-code platvormid, nende võimalused, tööriistad, eelised ja puudused. Olemasolev mobiilirakendus "MyStudy" on realiseeritud kasutades Mobincube no-code platvormi. Töös jõuti järeldusele, et Mobincube osutus sobivaimaks platvormiks eelkõige tänu oma lihtsusele mitte IT-inimese jaoks ning funktsionaalsusele, mis hästi sobis rakenduse realiseerimiseks.

Töö tulemusena on välja pakutud õpetaja ja spetsialisti poolt loodud mobiilirakendus reeglite õppimiseks algklasside õpilastele. Uue tarkvara kasutuselevõtt vähendas reeglite printimisele kuluvat aega ja paberit ning suurendas reeglite õppimisprotsessi efektiivsust.

Töö eesmärk on saavutatud. Õpetaja realiseeris kavandatud tarkvara ning on saanud mitteprofessionaalset tarkvaraarendajaks (sama tüüpi rakenduste tegemisel). Loodud rakenduses on palju võimalusi edasi arendamiseks.



## Kasutatud kirjandus

- [1] Põhikooli- ja gümnaasiumiseadus. [WWW]  
<https://www.riigiteataja.ee/akt/13337919> (18.03.2018)
- [2] Sillamäe Kannuka Kooli veebisait. [WWW] <http://www.kannukakool.edu.ee>  
(18.03.18)
- [3] Sillamäe Kannuka Kool. Ainevaldkonnad põhikoolis. [WWW]  
<http://kannukakool.edu.ee/rus/info/documents/edu/2015-programs/SPO-2015-format.pdf> (18.03.18)
- [4] Voomets, G. (2003). Bakalaureusetöö, «Erinevad õpetamismeetodid ja nende kasutamisest võimlemiselementide õpetamisel». [WWW]  
[http://dspace.ut.ee/bitstream/handle/10062/31255/gert\\_voomets2013.pdf?sequence=1](http://dspace.ut.ee/bitstream/handle/10062/31255/gert_voomets2013.pdf?sequence=1)  
(18.03.18)
- [5] SearchSoftwareQuality. Take the pain out of software developer hiring. [WWW]  
<https://searchsoftwarequality.techtarget.com/opinion/Take-the-pain-out-of-software-developer-hiring> (02.04.18)
- [6] Postimees ärileht. IT-teenuste hind pole varsti eestlasele tasukohane. [WWW]  
<https://majandus24.postimees.ee/4081609/it-teenuste-hind-pole-varsti-eestlasele-taskukohane> (02.04.18)
- [7] IT Koolitus. Programmeerimise kursused. [WWW]  
<https://www.koolitus.ee/et/uudised/20-viope-programmeerimise-kursused-java-php-c-sql-python> (07.05.18)
- [8] BCS Koolitus. [WWW] <http://www.bcskoolitus.ee/> (07.05.18)
- [9] Statcounter. Mobile Operating System Market Share World. [WWW]  
<http://gs.statcounter.com/os-market-share/mobile/world> (25.04.18)
- [10] Masterofcode. App Store vs Google Play: Stores in Numbers. [WWW]  
<https://masterofcode.com/blog/app-store-vs-google-play> (26.04.18)
- [11] Bizagi. Bizagi Modeler Documentation. [WWW] <http://help.bizagi.com/process-modeler/en/> (18.04.18)
- [12] LPgenerator. Глава 8. Путеводитель по человеко-компьютерному взаимодействию: разработка с участием конечных пользователей. [WWW]  
<https://lpgenerator.ru/blog/2017/09/30/putevoditel-po-cheloveko-kompyuternomu-vzaimodejstviyu-razrabotka-s-uchastiem-konechnyh-polzovatelej/> (05.05.18)

- [13] Hindawi. End User Development: Survey of an Emerging Field for Empowering People. [WWW] <https://www.hindawi.com/journals/isrn/2013/532659/> (05.05.18)
- [14] SpringerLink. End-User Development: An Emerging Paradigm. [WWW] [https://link.springer.com/chapter/10.1007%2F1-4020-5386-X\\_1](https://link.springer.com/chapter/10.1007%2F1-4020-5386-X_1) (06.05.18)
- [15] Wikipedia. Kasutaja: KJaagup/Kasutaja [WWW] <https://et.wikipedia.org/wiki/Kasutaja:KJaagup/Kasutaja> (06.05.18)
- [16] SearchSoftwareQuality. The the pain out of software developer hiring. [WWW] <https://searchsoftwarequality.techtarget.com/opinion/Take-the-pain-out-of-software-developer-hiring> (28.04.18)
- [17] SearchSoftwareQuality. How a low-code platform can end application sprawl forever. [WWW] [https://searchsoftwarequality.techtarget.com/feature/How-a-low-code-platform-can-end-application-sprawl-forever?utm\\_medium=EM&asrc=EM\\_NLN\\_86631413&utm\\_campaign=20171211\\_searchOracle%2520%253E%2520Cloud%2520Applications%2520Digest&utm\\_source=NLN&track=NL-1839&ad=918125&src=918125](https://searchsoftwarequality.techtarget.com/feature/How-a-low-code-platform-can-end-application-sprawl-forever?utm_medium=EM&asrc=EM_NLN_86631413&utm_campaign=20171211_searchOracle%2520%253E%2520Cloud%2520Applications%2520Digest&utm_source=NLN&track=NL-1839&ad=918125&src=918125) (28.04.18)
- [18] Skuid. The difference between no-code and low-code development. [WWW] <https://www.skuid.com/platform/why-skuid/no-code-low-code/> (29.04.18)
- [19] Forbes. The Low-Code/No-Code Movement: More Disruptive Than You Realize. [WWW] <https://www.forbes.com/sites/jasonbloomberg/2017/07/20/the-low-codeno-code-movement-more-disruptive-than-you-realize/#6d4f77ff722a> (29.04.18)
- [20] OutSystems. No Boundaries! Mobile Development Low-Code. [WWW] <https://www.outsystems.com/1/low-code-development-platforms-wave/> (03.05.18)
- [21] TestingFreak. Best 15 Platforms to Develop Mobile Apps Online for Android, iOS and Windows. [WWW] <http://testingfreak.com/best-15-platforms-develop-mobile-apps-online-android-ios-windows/> (29.04.18)
- [22] Techcrunch. Please don't learn to code. [WWW] <https://techcrunch.com/2016/05/10/please-dont-learn-to-code/> (03.05.2018)
- [23] Mobincube. [WWW] <https://www.mobincube.com/ru/>
- [24] Fischer, G., Giaccardi, E., Ye, Y., Sutcliffe, A.G. and Mehandjiev, N., "Meta-Design: A Manifesto for End-User Development." (08.09.18)
- [25] Wikipedia. Mobiilirakendus. [WWW] <https://et.wikipedia.org/wiki/Mobiilirakendus> (25.04.18)

## Lisa 1 - Küsitlus

Опрос для учеников:

1. Вам понравилось мобильное приложение, где собраны все правила за начальную школу? **Да/Нет**
2. Будете ли вы пользоваться данным приложением? **Да/Нет**
3. Удобно ли оно для пользования? **Да/Нет**

Küsitlus õpilastele:

1. Kas Teile meeldis mobiilirakendus, kus on kogutud kõik reeglid algkooli õpilaste jaoks? **Jah/Ei**
2. Kas tahaksite kasutada seda rakendust edasi? **Jah/Ei**
3. Kas antud rakendus on mugav kasutamisel? **Jah/Ei**

Опрос для учителей:

1. Возникла ли когда-нибудь надобность в подобном приложении? **Да/Нет**
2. Удобно для Вас данное приложение? **Да/Нет**
3. В чем плюс данного приложения для Вас? (отметьте подходящий(-ие) вариант(-ы) ответа(-ов)):
  - а) экономия времени на распечатывание и вклеивание правил;
  - б) все правила за начальную школу собраны в одном месте;
  - в) возможность использования приложения учениками вне школы;
  - г) свой вариант:.....
4. Хотели бы Вы использовать данное приложение для своего класса? **Да/Нет**
5. Оцените приложение по шкале от 1-5 (1- очень плохо, 2 - плохо, 3 - средне, 4 - хорошо, 5 - отлично). Подчеркните подходящий вариант ответа.

Küsitlus õpetajatele:

1. Kas tekkis kunagi vajadus niisugusest mobiilirakendusest? **Jah/Ei**

2. Kas antud rakendus on mugav Teie jaoks? **Jah/Ei**
3. Millised on antud rakenduse plussid Teie jaoks? (märkige sobiv(-vad) vastus(-sed)):
  - e) printimiseks ja kleepimiseks aega säästmine
  - f) kõik reeglid algkooli klasside jaoks on ühes kohas
  - g) võimalus kasutada rakendus kooli väljas
  - h) oma variant:

- .....
4. Kas Teie tahaksite kasutada antud mobiilirakendust oma klassi jaoks? **Jah/Ei**
  5. Hinnake rakendust 5-palli skaala põhjal ( 1- väga halb, 2 – halb, 3 – keskmine, 4 – hea, 5 – väga hea). Tõmmake sobiv variant alla.

## Lisa 2 – Küsitluse vastused

- ФРИАНТ
1. Вам понравилось мобильное приложение, где собраны все правила за начальную школу? Да/Нет
  2. Будете ли вы пользоваться данным приложением? Да/Нет
  3. Удобно ли оно для пользования? Да/Нет

1. Вам понравилось мобильное приложение, где собраны все правила за начальную школу? Да/Нет *да*
2. Будете ли вы пользоваться данным приложением? Да/Нет *да*
3. Удобно ли оно для пользования? Да/Нет *да*

1. Вам понравилось мобильное приложение, где собраны все правила за начальную школу? Да/Нет
2. Будете ли вы пользоваться данным приложением? Да/Нет
3. Удобно ли оно для пользования? Да/Нет

1. Вам понравилось мобильное приложение, где собраны все правила за начальную школу? Да/Нет
2. Будете ли вы пользоваться данным приложением? Да/Нет *Всегда*
3. Удобно ли оно для пользования? Да/Нет

1. Вам понравилось мобильное приложение, где собраны все правила за начальную школу? Да/Нет
2. Будете ли вы пользоваться данным приложением? Да/Нет
3. Удобно ли оно для пользования? Да/Нет

*АРТЕТ*

Joonis 17. Õpilaste vastused küsitlusele.

1. Возникла ли когда-нибудь надобность в подобном приложении? Да/Нет
2. Удобно для Вас данное приложение? Да/Нет
3. В чем плюс данного приложения для Вас? (отметьте подходящий(-ие) вариант(-ы) ответа(-ов):
  - а) экономия времени на распечатывание и вклеивание правил;
  - б) все правила за начальную школу собраны в одном месте;
  - в) возможность использования приложения учениками вне школы;
- г) свой вариант: .....
4. Хотели бы Вы использовать данное приложение для своего класса? Да/Нет
5. Оцените приложение по шкале от 1-5 (1- очень плохо, 2 - плохо, 3 - средне, 4 - хорошо, 5 - отлично). Подчеркните подходящий вариант ответа.

1. Возникла ли когда-нибудь надобность в подобном приложении? Да/Нет
2. Удобно для Вас данное приложение? Да/Нет
3. В чем плюс данного приложения для Вас? (отметьте подходящий(-ие) вариант(-ы) ответа(-ов):
  - а) экономия времени на распечатывание и вклеивание правил;
  - б) все правила за начальную школу собраны в одном месте;
  - в) возможность использования приложения учениками вне школы;
- г) свой вариант: экономия средств (кумага красна)
4. Хотели бы Вы использовать данное приложение для своего класса? Да/Нет
5. Оцените приложение по шкале от 1-5 (1- очень плохо, 2 - плохо, 3 - средне, 4 - хорошо, 5 - отлично). Подчеркните подходящий вариант ответа.

1. Возникла ли когда-нибудь надобность в подобном приложении? Да/Нет
2. Удобно для Вас данное приложение? Да/Нет
3. В чем плюс данного приложения для Вас? (отметьте подходящий(-ие) вариант(-ы) ответа(-ов):
  - а) экономия времени на распечатывание и вклеивание правил;
  - б) все правила за начальную школу собраны в одном месте;
  - в) возможность использования приложения учениками вне школы;
- г) свой вариант: универсальность
4. Хотели бы Вы использовать данное приложение для своего класса? Да/Нет
5. Оцените приложение по шкале от 1-5 (1- очень плохо, 2 - плохо, 3 - средне, 4 - хорошо, 5 - отлично). Подчеркните подходящий вариант ответа.

Joonis 18. Õpetajate vastused küsitlusele.

## Lisa 3 – Kannuka Kooli direktori arvamus



---

### Sillamäe Kannuka Kool

Tallinna Tehnikaülikool  
Infotehnoloogia teaduskond

Meie 02.05.2018.a. nr 1-6/31

Arvamus

Tallinna Tehnikaülikooli üliõpilane Jelena Fjodorova esitas oma diplomitöö raames tarkvara arendusäppi mobiilirakendus "MyStudy", mida me peame kooli õppetöö jaoks huvitavaks ja kasulikuks.

Juri Nikitin  
direktor

---

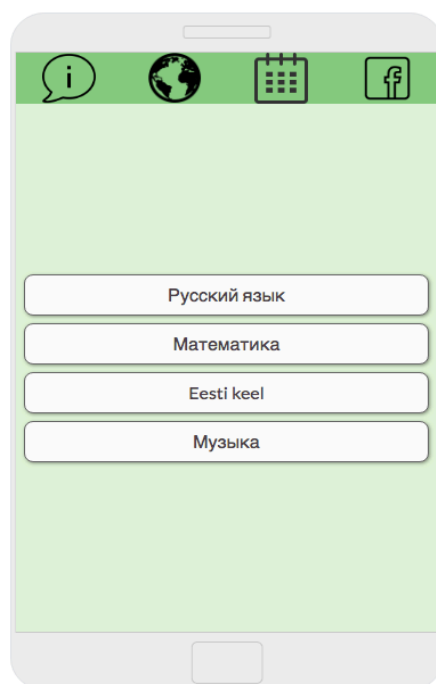
Geoloogia tn. 13,  
40233 Sillamäe

Tel. 3974044, fax. 3974044  
kannukakool@hotmail.ee

a/a 1120046943  
Hansapank

Joonis 19. Kannuka Kooli direktori arvamus loodud rakenduse kohta.

## Lisa 4 – Mobiilirakenduse ekraanipildid



Joonis 20. MyStudy rakenduse pealehelül.



Joonis 21. MyStudy rakenduse eesti keele lehekül.





Joonis 22. MyStudy rakenduse "Arvsõnad" teema reeglite lehekülg.