

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Renar Tupits 185003IADB

**Arenguestluste läbiviimist toetava tarkvara
loomine Eesti IT-ettevõtete näitel**

Bakalaureusetöö

Juhendaja Kristiina Hakk

PhD

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: Renar Tupits

13.02.2021

Annotatsioon

Arenguvestlused on eesmärgipärased ja süsteemsed vestlused, mis toimuvad üldjuhul juhi ja töötaja vahel ning on olulisel kohal töötajate töö tulemuslikkuse tõstmisel. Arenguvestluste läbiviimine on paljude juhtide jaoks keerukas tegevus, mis vajab vastavaid oskuseid ja aega nii selle ettevalmistamiseks, läbiviimiseks kui ka tulemuste analüüsimiseks. Seetõttu viiakse sageli ettevõtetes läbi mitteametlikke vestluseid, mis on ebakonkreetsed ning vähem tulemuslikumad.

Käesoleva bakalaureusetöö eesmärk on luua Eesti IT-ettevõtete ootustele vastav struktureeritud ja süsteemsust tagav arenguvestluste läbiviimist toetav tarkvara. Lõputöö eesmärgi saavutamiseks uuriti Eesti IT-ettevõtete ootuseid loodava tarkvara kohta. Saadud tulemused prioritseeriti ning selgitati välja peamised funktsionaalsed ja mittefunktsionaalsed nõuded. Valmistati rakenduse prototüüp ning kirjeldati loodava rakenduse valmimisel kasutatud tehnoloogiaid.

Eesti IT-ettevõtete seas läbiviidud küsitlusest selgus, et loodav rakendus võiks leida kasutust pea kõikides uuringus osalenud ettevõtetes. Lõputöö tulemusena valmis rakenduse esmane versioon, mis võimaldab rakendusse lisada uusi töötajaid, koostada ettevõttesiseseid meeskondi ning arenguvestluseid läbi viia nii vabas vormis kui ka kasutades varasemalt koostatud küsitluse malle.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 37 leheküljel, 8 peatükki, 18 joonist ja 6 tabelit.

Abstract

Development of an Application Supporting the Performance Appraisal Interviews on the Example of Estonian IT Companies

Performance appraisal interviews are purposeful and systematic interviews which usually take place between the manager and the employee and play an important role in improving the efficiency of the employee's work. Conducting performance appraisal interviews is a complex activity for many managers, which requires the appropriate skills and time to prepare, conduct and analyse the results. As a result, companies often conduct informal interviews that are vague and less effective.

The aim of this bachelor's thesis is to create a structured and systematic software that supports conducting of performance appraisal interviews and meets the expectations of Estonian IT companies. In order to achieve the goal of the thesis, the expectations of Estonian IT companies regarding the software to be created were found. The obtained results were prioritized, and the main functional and non-functional requirements were identified. A prototype of the application was prepared, and the technologies used in the completion of the application were described.

The survey conducted among Estonian IT companies revealed that the created application could be used in almost all companies participating in the survey. As a result of the thesis, the initial version of the application was developed. It allows adding new employees to the application, forming companies' internal teams and conducting performance appraisal interviews both in free form and using previously prepared survey templates.

The thesis is in Estonian and contains 37 pages of text, 8 chapters, 18 figures and 6 tables.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , rakendusliides
CSS	<i>Cascading Style Sheets</i> , kaskaadlaadistik
DOM	<i>Document Object Model</i> , dokumendi objektimudel
GCP	<i>Google Cloud Platform</i> , pilveandmetöötluse platvorm
HTML	<i>HyperText Markup Language</i> , hüperteksti märgistuskeel
HTTPS	<i>Hypertext Transfer Protocol Secure</i> , protokoll andmete turvaliseks edastamiseks
JSX	<i>JavaScript XML</i> . ReactJS kasutatav süntaks, mis on javascripti laiendus
NoSQL	<i>Not Only Scripted Query Language</i> , mitterelatsiooniline alternatiivne andmebaas
PaaS	<i>Platform as a Service</i> , teenusepakkuja poolt hallatav platvorm.
POM	<i>Project Object Model</i> , projekti informatsioonist koosnev fail
SPA	<i>Single Page Application</i> , üheleheline rakendus
SQL	<i>Structured Query Language</i> , relatsioonilistele andmebaasidele mõeldud päringukeel
XML	<i>Extensible Markup Language</i> , laiendatav märgistuskeel

Sisukord

1	Sissejuhatus.....	10
2	Arenguvestlus ja selle ülesehitus	12
3	Ülesandepüstitus	15
3.1	Probleem	15
3.2	Metoodika	15
4	Uuringu tulemused ja rakenduse analüüs	17
4.1.1	Vastajate taust	17
4.1.2	Ootused loodavale rakendusele.....	19
4.2	Olemaolevad lahendused.....	21
4.3	Rakenduse analüüs.....	22
4.3.1	Funktsionaalsed nõuded loodavale rakendusele	22
4.3.2	Mittefunktsionaalsed nõuded	23
4.3.3	Funktsionaalsete nõuete prioritseerimine MoSCoW mudeliga	23
4.3.4	Planeeritava rakenduse prototüüp	25
4.3.5	Planeeritava rakenduse prototüübi joonised	26
4.3.6	Kasutusmallid	30
5	Kasutatud tehnoloogiad ja raamistikud.....	33
5.1	Rakenduse majutamine	33
5.2	Rakenduse serveri-poolne kirjeldus	35
5.2.1	Kotlin	36
5.2.2	Spring Boot	36
5.2.3	Maven	37
5.3	Datastore	38
5.4	Rakenduse kliendi-poolsed tehnoloogiad	39
6	Rakenduse realisatsioon.....	40
6.1	Serverirakendus.....	40
6.2	Rakenduse API lõpp-punktid	42

6.3	Rakenduse kasutajate rollid	43
6.4	Kasutaja lisamine, meeskonna loomine ja õiguste jagamine	43
6.5	Arenguestluste läbiviimine	44
7	Tuleviku arendustegevused.....	46
8	Kokkuvõte.....	47
	Kasutatud kirjandus	48
	Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks.....	51
	Lisa 2 – Küsimustik IT-ettevõtetele.....	52

Jooniste loetelu

Joonis 1. Intervjuu etapid [2].	13
Joonis 2. Küsitluses osalenud ettevõtete suurus töötajate arvu järgi.	17
Joonis 3. Küsitluses osalenud ettevõtete vanus tegutsemisaastate järgi.	18
Joonis 4. Küsitluses osalenud ettevõtete arenguestluse tarkvara kasutamine.....	19
Joonis 5. Prototüübi ekraanivaade uue isiku lisamisel.....	26
Joonis 6. Prototüübi ekraanivaade meeskonna koosseisust.	27
Joonis 7. Prototüübi ekraanivaade arenguestluse alustamisel.....	27
Joonis 8. Prototüübi ekraanivaade vabas vormis arenguestluse läbiviimisel.	28
Joonis 9. Prototüübi ekraanivaade arenguestluse läbiviimisel kasutades küsimustiku.....	28
Joonis 10. Prototüübi ekraanivaade uue küsimustiku koostamisel.....	29
Joonis 11. Prototüübi ekraanivaade varasemalt läbiviidud arenguestlustest.	29
Joonis 12. Google Cloud Platform töölaud.....	34
Joonis 13. Rakenduse konfiguratsioon GCP AppEngine jaoks.	35
Joonis 14. Minimaalsed nõuded POM jaoks.....	37
Joonis 15. Serveri-poolne projekti struktuur.....	40
Joonis 16. Kontrolleri klassi meetod kasutaja profiili küsimiseks.....	41
Joonis 17. Kuvatõmmis rakendusest uue kasutaja lisamisel.....	44
Joonis 18. Arenguestluste läbiviimise jadaskeem.....	45

Tabelite loetelu

Tabel 1. Loodava tarkvara funktsionaalsete nõuete kirjeldavad statistikud.	20
Tabel 2. Funktsionaalsete nõuete prioritseerimine MoSCoW mudeli abil.	24
Tabel 3. Kasutusmall „Uue isiku lisamine“.....	30
Tabel 4. Kasutusmall „Arenguvestluse malli koostamine“.....	31
Tabel 5. Kasutusmall „Uue arenguestluse alustamine“.....	32
Tabel 6. Rakenduse lõpp-punktide loetelu.....	42

1 Sissejuhatus

On teada, et organisatsioonide tulemuslikkus sõltub eelkõige töötajatest ning nende töö tulemuslikkusest. Töötajate arengu toetamiseks on oluline inimressursside juhtimisele keskenduda strateegiliselt. Üheks kriitilise tähtsusega tegevuseks on töötajate töösoorituse hindamine, tänu millele on võimalik mõista personali arengu- ja koolitusvajadusi, motivatsiooni ning töö tulemusi. Uuemad lähenemised töösoorituse hindamisel ei keskendu mitte niivõrd töötajate tegevustele ja tulemustele vaid organisatsiooni ja töötajate arendamisele. Eelkõige keskendutakse indiviidi ettevõtlikkusele, õppimise tõhususele ning tulevikule suunatud näitajatele.

Üheks oluliseks töösoorituse hindamise meetodiks on arenguestlused. Arenguestlused on eesmärgipärased ja süsteemsed vestlused, mis toimuvad üldjuhul juhi ja töötaja vahel. Selle käigus toimub töötaja hetkeolukorra ja varasemate eesmärkide saavutatuse hindamine, uute eesmärkide seadmine ning tegevuskava loomine, andes võimaluse mõlemale osapoolle ettepanekute tegemiseks ning avatud vestluseks.

Arenguestluste tõhusus pikemas perspektiivis sõltub süsteemsustest ja järjepidevustest. Selle järgselt on oluline teha kokkuvõtte, kus kajastuvad tehtud tegevused ning saavutused, aga ka tuleviku eesmärgid, tööülesanded ning vastutused.

Eestis on hetkel teada kaks tarkvaralahendust (PlanPro, Upsteem), mis pakuvad arenguestluste läbiviimiseks ning töötaja arengu kaardistamiseks võimalusi eelkõige avalikule sektorile. Leidub mitmeid uurimusi, mille käigus on konkreetsete Eesti ettevõtete näitel kaardistatud arenguestluste efektiivsust ning välja töötatud organisatsioonisiseseid tarkvarasid, kuid IT sektoris puudub universaalne ning kasutajate ootustele vastav tarkvaralahendus.

Bakalaureusetöö eesmärk on luua Eesti IT-ettevõtete ootustele vastav struktureeritud ja süsteemsust tagav arenguestluste läbiviimist toetav tarkvara.

Lõputöö koosneb kaheksast peatükist, sh sissejuhatus ja kokkuvõte. Teises peatükis kirjeldab kirjeldab töö autor arenguestluste ülesehitust ning tutvustab, millest koosneb struktureeritud ja süsteemne vestlus. Töö kolmandas osas kirjeldatakse probleemi ning tehakse ülevaade meetodikast. Töö neljandas peatükis on tehtud ülevaade uuringus osalenutest ning luuakse tabel, kus on uuringus osalenute hinnangud loodava tarkvara funktsionaalsetele nõuetele. Lisaks loetletakse rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded, mis prioriseeritakse MoSCoW mudeliga. Analüüsi käigus valmib rakenduse prototüüp ning kirjeldatakse kolme kasutusmalli. Peatüki lõpus teeb autor põgusa ülevaate olemasolevatest lahendustest. Töö viiendas osas tutvustab autor rakenduse valmimisel kasutatud tehnoloogiaid ja raamistike. Kuuendas peatükis tutvustab autor valminud rakenduse ülesehitust ja põhiprotsesse ning töö seitsmendas peatükis annab ülevaate tuleviku arendustegevustest. Töö praktilise osana valmib arenguestluste läbiviimist toetav tarkvara.

2 Arenguestlus ja selle ülesehitus

Käesolevas peatükis annab töö autor ülevaate arenguestluste olemusest, eesmärkidest ja sisust. Teoreetiline ülevaade on sisendiks töö järgnevate etappide koostamisel.

Arenguestlus on eelnevalt ettevalmistatud vestlus, mis toimub üldjuhul juhi ja töötaja vahel. Vestluse eesmärk on üksteise parem mõistmine, juhi ja alluva töötahte suurendamine ning töömotiveerimise toetamine [1]. Õigesti läbi viidud arenguestlused annavad võimaluse töötajate arengust ülevaate saamiseks, selgete ja mõõdetavate eesmärkide püstitamiseks, töötajate vajaduste ja motiveerituse hindamiseks ning organisatsiooni koostöö ja arenemise kujundamiseks [2].

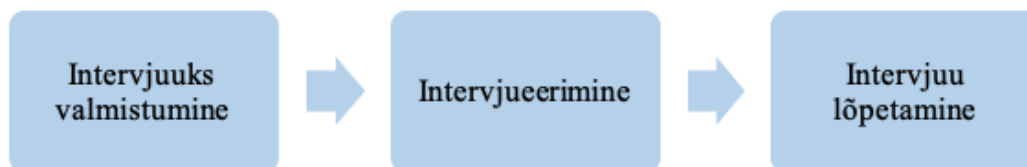
Merkuljeva on välja toonud arenguestluste konkreetsemad eesmärgid järgnevalt [1]:

1. Töö eesmärkidest lähtuvalt töötaja pädevuste hindamine eneseanalüüsi kaudu.
2. Eesmärkide seadmine ja kokkuleppimine eelseisvaks tööperioodiks.
3. Arenguplaani kokkuleppimine eelseisvaks tööperioodiks.
4. Nii töötaja kui ka juhi adekvaatse enesehinnangu kujunemise toetamine.

Arenguestlust võib käsitleda kui intervjuud, millel on oma eesmärk, valdkond ja vorm ning oma kavandatuse astmest lähtuvalt võib need jagada struktureeritud intervjuuks ning struktureerimata intervjuuks [2]. Varasematest uurimustest on selgunud, et Ameerika Ühendriikides kasutatakse vestluste läbiviimisel 70% juhtudest struktureerimata intervjuusid, kuid Euroopas on enam levinud struktureeritud intervjuud [3].

Struktureeritud intervjuu sarnaneb ankeedile, mille küsimused on varasemalt täpselt paika pandud ning neid esitatakse kindlas järjekorras. Antud intervjuu liik ei võimalda aga vestlust loomulikult suunata ning töötajale paindlikult läheneda. Struktureerimata intervjuu on oma ülesehituselt vaba, kuid selles keskendutakse olulisematele põhiteemadele. Sellise lähenemise puhul on aga oht, et saadud info fikseerimine ning hilisem hindamine võib olla probleemne, mistõttu Fisher, Schoenefeldt, Shaw [4] sõnul on kõige tulemuslikum kahe intervjuu liigi kombineerimine ehk poolstruktureeritud intervjuude läbiviimine.

Mistahes intervjuu puhul on aga oluline intervjuueerimise strateegia ja taktika valik ning selle rakendamine. Intervjuu puhul võib eristada kolme peamist etappi, mis on kujutatud alljärgneval joonisel (Joonis 1). Intervjuuks valmistumisel on oluline osa intervjuu kavandamisel, sh intervjuu liigi valimisel ning vestluse aja ja koha fikseerimisel. Intervjuueerimise käigus on oluline vestluse suunamine, sellele keskendumine ning ajaraamis püsimine. Intervjuu lõppedes toimub järelduste tegemine ning vestluste tulemuste fikseerimine.



Joonis 1. Intervjuu etapid [2].

Erinevate tasandite juhtide, allüksuste ning töötajate huvide tagamiseks soovitatakse arenguvestluste läbiviimisel alustada juhtivatest ametipositsioonidest liikudes sealt astmeliselt järgmiste ametipositsioonideni. Üldjuhul tegelevad personali hindamisega organisatsiooni juhid ja spetsialistid, kuid üha enam keskendutakse ka erinevate osapoolte hinnangutele, mistõttu on hakatud eristama ka erinevaid hindajaid. Need võivad olla vahetu juht, kolleeg, klient ning hinnatav ise [4]. Nimetatud osapoolte hindamist kasutatakse eelkõige uuemate 360-kraadiliste tagasisidestamiste puhul, kuid parema hindamistulemuse tagab hindajate rohkus mistahes meetodite rakendamisel. Seda kinnitavad ka uuemad lähenemised hindamisele tuues välja ka töötajate aktiivsust, avatust ja kaasatust hindamisprotsessis [2].

Kuna arenguvestluste ehk intervjuude keskmes on töötajate töösoorituse hindamine on üheks võimalikuks lahenduseks selle ülesehitamisel töösoorituse hindamise põhinäitajatele keskendumine.

Türk sõnul on töösoorituse hindamise põhinäitajateks [2]:

- töötulemused;
- tegevused ehk käitumine;
- töötaja pädevused ehk teadmised, oskused, kogemused ja võimed.

Just pädevuste ning tegevuste hindamisel on võimalik suunata töötajate arengut, koolitusvajadusi ning erinevate ametipositsioonide ümberpaigutamist. Mitmed varasemad uurimused on aga leidnud, et tööd iseloomustavad näitajad peavad olema seotud konkreetse töö iseloomu ning organisatsiooni tervikliku vaatega, mistõttu ei ole mõistlik universaalseid põhinäitajaid luua.

Ehkki mitmed olulised osad arenguestlustest sõltuvad konkreetsest kontekstist ning eesmärkidest, on teada, et mistahes vestluste liigi ja sisu puhul on kriitilise tähtsusega intervjuu lõpus kokkuvõtete tegemine. Selleks on vestluse eesmärkide ja kestuse fikseerimine, protokoll koostamine ning saadud info oskuslik kogumine (nt hinnangutel põhineva skaala kasutamine). Seda tehakse nii erinevate osapoolte selguse kui protsessi süsteemsuse tagamiseks, sest fikseerimata töötaja praegust hetkeolukorda ei ole võimalik töötaja arengut järgmisel vestlustel mõõta. Korrektne info kogumine ja talletamine annab võimaluse selle hilisemaks süstematiseerimiseks ning töötlemiseks.

Tuginedes teoreetilisele ülevaatele on tarkvaralahenduse loomisel võetud arvesse järgmist:

- Tarkvaralahendus peab koosnema kolmest etapist:
 - o intervjuu kavandamine (intervjuu liik, aeg ja koht);
 - o intervjuueerimine (kestvus, küsimustiku sisu);
 - o kokkuvõtete tegemine (eesmärkide püstitamine, kokkulepete sõlmimine, arenguplaani loomine).
- Tarkvaralahendus peab võimaldama intervjuu läbiviija rolli valikut.
- Tarkvaralahendus peab võimaldama nii struktureeritud kui ka struktureerimata intervjuude läbi viimist.
- Tarkvaralahendus peab keskenduma töösoorituse hindamise põhinäitajatele, kuid ei pea sisaldama konkreetseid töösoorituse hindamise indikaatoreid.
- Tarkvaralahendus peab võimaldama töötaja eneseanalüüsi.
- Tarkvaralahendus peab võimaldama vestluse fikseerimist (töötulemused, eesmärgid, kokkulepped jms).

3 Ülesandepüstitus

Käesolevas peatükis annab töö autor ülevaate lahendatavast probleemist ning lõputöö metoodikast.

3.1 Probleem

Arenguvestluste läbiviimine on paljude juhtide jaoks keerukas tegevus, mis vajab vastavaid oskuseid ja aega nii selle ettevalmistamiseks, läbiviimiseks kui ka tulemuste analüüsimiseks. Seetõttu viiakse sageli ettevõtetes läbi mitteametlikke vestluseid, mis on ebakonkreetsed ning vähem tulemuslikumad [2]. Arenguvestluste tõhusus pikemas perspektiivis sõltub süsteemsustest ja järjepidevustest. Selle järgselt on oluline teha kokkuvõte, kus kajastuvad tehtud tegevused ning saavutused, aga ka tuleviku eesmärgid, tööülesanded ning vastutused.

Eestis on hetkel teada kaks tarkvaralahendust (PlanPro, Upsteem), mis pakuvad arenguvestluste läbiviimiseks ning töötaja arengu kaardistamiseks võimalusi eelkõige avalikule sektorile. Leidub mitmeid uurimusi [5], [6], mille käigus on konkreetsete Eesti ettevõtete näitel kaardistatud arenguvestluste efektiivsust ning välja töötatud organisatsioonisiseseid tarkvarasid, kuid IT sektoris puudub universaalne ning kasutajate ootustele vastav tarkvaralahendus.

Lähtuvalt eeltoodust soovib lõputöö autor pakkuda lahendust struktureeritud arenguvestluste läbiviimiseks ning süsteemsuse tagamiseks IT valdkonnas.

3.2 Metoodika

Käesoleva uurimistöö tegevused on jaotatud kaheks osaks.

Töö analüüsi osas kogub töö autor arenguvestluste läbiviimist toetava tarkvara jaoks ärinõudeid Eesti IT-ettevõtetelt. Saadud info põhjal koostatakse ärianalüüs, mille käigus prioritseeritakse nõuded. Lisaks valmib rakenduse esmane prototüüp, kasutades selleks veebipõhiseid tehnoloogiaid. Autor annab ülevaate Eesti turul olevatest sarnastest lahendustest ning võrdleb neid loodava rakendusega.

Töö praktilise osana valmib koostatud analüüsi põhjal arenguveestluste läbiviimist toetav tarkvararakendus. Autor kirjeldab valminud rakenduse arhitektuuri ning loodud rakenduse põhilisi protsesse. Välja tuuakse võimaluse olemasoleva lahenduse edasiarendamiseks.

Uurimistöo analüüsi üks osa on kvantitatiivne uurimus, mille raames viidi läbi Google Forms keskkonnas veebipõhine küsimustik Eesti IT-ettevõtete seas. Küsimustik koostati lähtuvalt uurimistöo analüüsi osa eesmärgist ning see koosnes 18 küsimusest (Lisa 2). Esimene osa küsimustikust keskendus IT-ettevõtete taustainfole (nt ettevõtte suurus ja vanus) ning teine osa sisaldas ettevõtte hinnanguid loodava tarkvaralahenduse funktsionaalsetele nõuetele. Ootuseid loodavale tarkvaralahendusele koguti 25 väitega ning iga väite puhul paluti vastajatel tarkvaralahenduse funktsionaalsuse olulisust hinnata viie palli skaalal. Kõige madalam hinnang oli „Ei nõustu üldse“ ning kõige kõrgem hinnang oli „Nõustun täielikult“. Küsitluse vastamine oli vabatahtlik ning anonüümne. Töös esitatakse saadud tulemused statistilise ülevaadena. Kogutud andmeid talletatakse pilveteenuses ja arvutis.

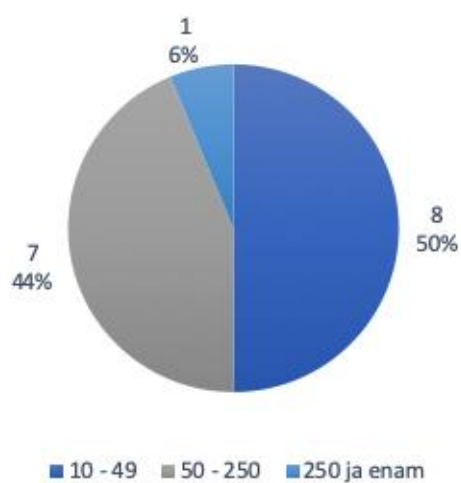
Uuringu valim oli eesmärgipärane valim, mille moodustamisel oli ainsaks kriteeriumiks ettevõtte tegevusvaldkond (IT sektor). Uuringukutse saadeti 47 Eesti IT-ettevõttele ajavahemikus 15.03.2021 – 18.04.2021. Selle ajaga koguti 16 vastust. Saadud tulemuste analüüsimiseks ja töötlemiseks kasutati Microsoft Excel tarkvara. Küsimustikust saadud andmed kodeeriti asendades kõik küsimustiku hinnangud numbrilise väärtusega. Hinnang „Ei nõustu üldse“ asendati väärtusega „1“, hinnang „Ei nõustu“ väärtusega „2“, hinnang „Nii ja naa“ väärtusega „3“, hinnang „Nõustun“ väärtusega „4“ ning hinnang „Nõustun täielikult“ väärtusega „5“. Uurimusest saadud andmed on esitatud töö tulemuste peatükis.

4 Uuringu tulemused ja rakenduse analüüs

Järgnevalt on antud ülevaade Eesti IT-ettevõtete seas läbi viidud küsitluse tulemustest. Kogutud andmed on aluseks rakenduse nõuete loomisel.

4.1.1 Vastajate taust

Küsitlusele vastanud ettevõtted (n=16) jagunesid ettevõtte suuruse ehk töötajate arvu järgi peaaegu võrdselt (Joonis 2). Vaid ühes ettevõttes oli 250 ja enam töötajat. Küsitlusele vastanud ettevõtetest ei olnud üheski ettevõttes alla kümne töötaja.

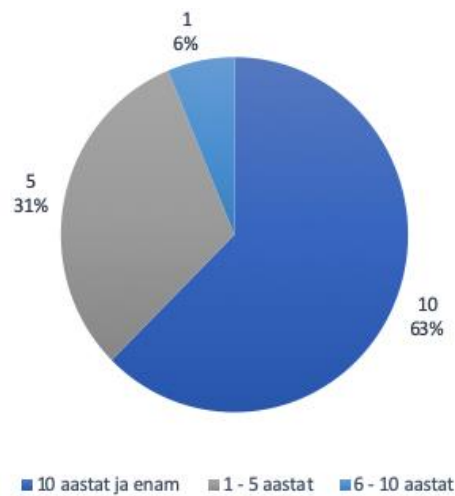


Joonis 2. Küsitluses osalenud ettevõtete suurus töötajate arvu järgi.

Üle poole ettevõtetest (n=10) olid tegutsenud kümme ja enam aastat. Küsitlusele vastanud ettevõtetest ei olnud mitte ükski tegutsenud alla aasta (Joonis 3).

Kõik uuringus osalevad ettevõtted pidasid oluliseks töötajate arengu toetamist ning neis viiakse töötajatega läbi arenguestluseid. Valdaval osal ettevõtetest (n=13) on olemas kindel kord või struktuur arenguestluste läbi viimiseks. Vaid kahel ettevõttel puudub selline kord ning ühel

juhul sõltub arenguestluste läbiviimine tiimijuhist ning seetõttu on ka struktuur tiimipõhine. Üldine rahulolu arenguestluste süsteemi/korraga enda ettevõttes oli viie palli skaalal 3,68.

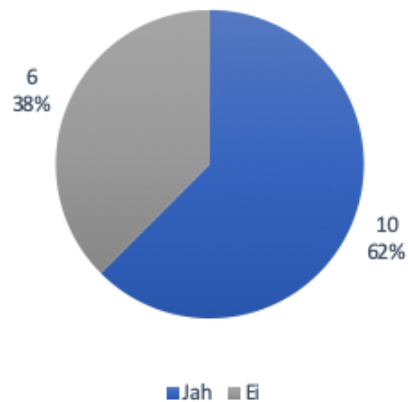


Joonis 3. Küsitluses osalenud ettevõtete vanus tegusemisaastate järgi.

Kõikides ettevõtetes (n=16) viib arenguestlusti tavapäraselt läbi tiimijuht. Ühel juhul viib arenguestlusti läbi vanemarendaja ning ühel juhul saab iga meeskonnaliige ise valida isiku, kellega vestluseid pidada. Täiendavalt toodi ühel juhul välja, et arenguestluste eelneb küsimustik, mille töötaja täidab. Vestluse käigus käsitletud teemade ja seatud eesmärkide salvestamine toimub üle pooltes ettevõtetes (n=9) mõnes pilveteenuses. Kuues ettevõttes toimub andmete salvestamine arvutis ning ühel juhul siseveebis.

Arenguestluste tõhususe/efektiivsuse mõõtmist käsitlevas avatud küsimusest (n=14) selgus, et pooltes ettevõtetes (n=7) sellist hindamist ei toimu. Teised ettevõtted tõid välja eelkõige töötajate rahulolu-uuringu, kus on võimalik hinnata enesearengut ja võimalusi. Lisaks sellele lähtutakse hindamisel ka vestluste tagasisidest, toimumise sagedusest, vaatlusest ning inimeste arengust.

Üle poole uuringus osalevatest ettevõtetest kasutab arenguestluste läbiviimiseks mõnda tarkvara (Joonis 4). Kasutatavate tarkvaradena tõid ettevõtted (n=11) välja tarkvarad Clanbeat (n=2), Lattice (n=2), Google lahendused (n=2), Bamboo HR (n=1), Notion (n=1) ning majasisesed lahendused (n=3).



Joonis 4. Küsitluses osalenud ettevõtete arengueestluse tarkvara kasutamine.

Ettevõttes kasutatava lahendusega olid täielikult rahul kolm ettevõtet, ülejäänud ettevõtted (n=8) leidsid, et kasutataval tarkvaral esinevad mõned puudused, kuid nad on sellega üldiselt rahul.

4.1.2 Ootused loodavale rakendusele

Järgnevas tabelis (Tabel 1) on välja toodud uuringus osalenud IT-ettevõtete (n=16) hinnangud loodava tarkvara funktsionaalsetele nõuetele. Tabelis on esitatud kõikide väidete aritmeetilised keskmised, standardhälbed ning miinimum ja maksimum hinnang väitele.

Tabel 1. Loodava tarkvara funktsionaalsete nõuete kirjeldavad statistikud.

Väide	M	SD	Min	Max
Rakendus peab omama töölauda, kus on statistika ettevõtte töötajate kohta.	3,68	0,87	2	5
Rakendus peab võimaldama integratsiooni personalihaldus tarkvaraga.	3,81	1,42	1	5
Rakendus peab võimaldama kasutajate importimist rakendusse.	3,68	1,35	1	5
Rakendus peab võimaldama meeskondade loomist olemasolevatest kasutajatest.	4,18	0,91	2	5
Rakendus peab võimaldama ettevõtte haldaja kontoga õiguste andmist teistele kasutajatele.	4,31	0,94	2	5
Rakendus peab võimaldama kasutajatele saata teateid ja meeldetuletusi.	4,5	0,73	3	5
Rakendus peab võimaldama töötajatel anda tagasisidet ettevõttele.	4,31	0,87	2	5
Rakendus peab võimaldama töötajatel teha ettepanekuid ettevõttele.	4,25	0,93	2	5
Rakendus peab võimaldama ettevõtte uudisvoo lisamist.	2,5	1,03	1	5
Rakendus peab võimaldama töökeele vahetamist inglise keelde.	4,75	0,44	4	5
Rakendus peab võimaldama töökeele vahetamist vene keelde.	2,37	1,31	1	5
Rakendus peab olema kasutajasõbralik vaegnägijatele ja kuulmispuudega inimestele.	3,5	1,36	1	5
Rakendust peab olema võimalik kasutada võrguühenduseta.	2,37	1,31	1	5
Rakendus peab võimaldama sõnumivahetust teiste kasutajatega.	2,62	1,36	1	5
Rakendus peab võimaldama valida erinevat tüüpi arenguveestluste vahel.	3,87	1,08	2	5
Rakendus peab võimaldama koostada malle arenguveestluste läbiviimiseks.	4,44	0,72	3	5
Rakendus peab võimaldama arenguveestluse läbiviijal lisada arenguveestluse käigus personaalseid märkmeid.	4,56	0,62	3	5
Rakendus peab võimaldama ekspordi varasemalt läbiviidud arenguveestlustele (failiformaatiks json, csv).	3,87	1,36	1	5
Rakendus peab võimaldama kasutajal näha arenguveestluste käigus sõlmitud kokkuleppeid.	4,56	0,62	3	5
Rakendus peab võimaldama defineerida ettevõttesiseseid rolle (arendaja, testija, jt).	4,06	0,77	3	5
Rakendus peab võimaldama rollidele lisada erinevaid tasemeid (junior, mid, senior, jt).	3,93	0,92	3	5
Rakendus peab võimaldama kasutajal muuta temaga sõlmitud kokkulepete staatust.	4,06	0,92	2	5
Rakendus peab võimaldama kasutajal näha enda arenguplaani.	4,43	0,72	3	5
Rakendus peab võimaldama kasutajal muuta enda profiili.	4,18	0,83	3	5
Rakendus peab võimaldama arenguveestluse läbiviijal määrata, mida teine osapool arenguveestluse protokollist näha saab.	4	0,89	3	5

Märkused. M – aritmeetiline keskmine, SD – standardhälve, Min – minimaalne, Max – maksimaalne.

4.2 Olemasolevad lahendused

Töö autor uuris internetist olemasolevaid lahendusi, mis toetavad arenguvestluste läbiviimist ning leidis kaks tarkvaralahendust, mis on kasutusel Eestis ja loodud Eesti ettevõtete poolt. Antud peatükis tutvustab autor põgusalt neid tarkvarasid ning lisaks ka ühte üldisemat lahendust, mida saab kasutada arenguvestluste läbiviimiseks.

PlanPro [10] on ettevõtte/rakendus, mis tegeleb veebipõhise juhtimistarkvara arendamise ja konsultatsiooniteenuse pakkumisega. PlanPro kliendibaas on väga lai, mille peamiseks kasutajateks on suur osa Eesti ministeeriumeid ja teisi avaliku sektori asutusi. Lisaks tehakse koostööd ka Moldova ja Montenegro valitsusega.

Selle tugevusteks võib pidada *Active Directory* või *Office 365* liidese abil kasutajate süsteemi lisamine, mis tagab ka olemasolevad rollid ning arenguvestlust alustades eristab juhi ja hinnatava. Lisaks on PlanPro veebilehel näha, et toimub pidev arendus uue funktsionaalsuse näol ning olemasolevate vigade parandamine. Kuna autoril puudub ligipääs kasutamaks PlanPro tarkvara, ei ole võimalik hinnata rakenduse funktsionaalsust. Samuti ei õnnestunud autoril saada infot rakenduse maksumuse kohta.

Upsteem.com [11] on 2011 aastal loodud ettevõtte, mille eesmärgiks on aidata ettevõtetel läbi viia arenguvestlusi, 360° tagasiside ja rahulolu-uuringuid. Veebilehel on näha klientideks suuremad ettevõtted Eestis, sh Statoil, Ragn-Sells ja Maanteeamet. Sarnaselt eelnevalt kirjeldatud rakendusele, puudub töö autoril ligipääs ka sellele rakendusele, mistõttu pole võimalik sellega lähemalt tutvuda ning võrrelda loodava rakendusega. Samuti puudub info rakenduse hinnastamise kohta. Vaadates Upsteem veebilehe avalehel olevat umbes poole minutit video, saab aimu, et tarkvaral on mõningaid ühiseid funktsionaalsusi käesoleva lõputöö raames planeeritava rakendusega. Võimalik on koostada küsimustike/malle ning neid kasutada vestluse käigus, teha märkmeid ja kommentaare.

Google Sheets [12] on veebipõhine kontoritarkvarapaketti, mille abil on samuti võimalik arenguvestlusi läbi viia. Selle eeliseks on töökindlus, kuna rakenduse arenduse eest vastutavad mitmed Microsofti meeskonnad ja selle kasutamine ei maksa ettevõttele midagi. Küll aga Google Sheets ei ole mõeldud otseselt arenguvestluste läbiviimiseks ning varem või hiljem võivad tabada teatud probleemid kasutamaks seda lahendust. Probleemiks võiks olla näiteks vestluste haldamine. Ettevõtte personali haldaval inimesel võib olla üks suur dokument, kus on

ülevaade töötajate arenguestlustest, temaga seatud eesmärkidest ja kokkulepetest. Juhul kui on kasutusel üks suur dokument, ei ole võimalik antud lahendust kasutades jagada seda töötajaga, ilma et näha poleks teiste inimeste isikliku infot, mis omakorda tähendab, et töötajal, kellega arenguestlust läbi viiakse, puudub võimalus näha arenguestluste ajalugu, protokolle ning kokkuleppeid. Üks võimalus oleks teha igale töötajale eraldi dokument ning seda vaid temaga jagada, kuid see lahendus sobiks vaid pigem väikestele ettevõtetele, kuna sadade dokumentide haldamine võib osutuda ajakulukaks ning segadust tekitavaks.

4.3 Rakenduse analüüs

Selles peatükis kirjeldab töö autor funktsionaalseid ja mittefunktsionaalseid nõudeid, millest funktsionaalsed nõuded prioritseeritakse MoSCoW mudeliga. Lisaks tutvustatakse valminud prototüüpi ning kirjeldatakse kolme kasutusmalli.

4.3.1 Funktsionaalsed nõuded loodavale rakendusele

Funktsionaalsete nõuete kogumine on oluline etapp tarkvaraarenduse protsessis. See annab hea ülevaate juba varases etapis, millele loodav rakendus hakkab tuginema ning koostöös kliendiga saab kaardistada tegevused, mis rahuldaksid kliendi vajadusi. Kokkuvõtlikult võib öelda, et funktsionaalsed nõuded on süsteemi poolt pakutavad tegevused ja teenused. Alljärgnevalt on välja toodud funktsionaalsed nõuded loodavale rakendusele.

Rakendus peab omama töölauda, kus on statistika ettevõtte töötajate kohta ja võimaldama:

- integratsiooni personalihaldus tarkvaraga;
- kasutajate importimist rakendusse;
- meeskondade loomist olemasolevatest kasutajatest;
- ettevõtte haldaja kontoga õiguste andmist teistele kasutajatele;
- kasutajatele saata teateid ja meeldetuletusi;
- töötajatel anda tagasisidet ettevõttele;
- töötajatel teha ettepanekuid ettevõttele;
- ettevõtte uudisvoo lisamist;
- töökeele vahetamist inglise keelde;
- töökeele vahetamist vene keelde;
- sõnumivahetust teiste kasutajatega;

- valida erinevat tüüpi arenguestluste vahel;
- koostada malle arenguestluste läbiviimiseks;
- võimaldama arenguestluse läbiviijal lisada arenguestluse käigus personaalseid märkmeid;
- ekspordi varasemalt läbiviidud arenguestlustele (failiformaatiks json, csv);
- kasutajal näha arenguestluste käigus sõlmitud kokkuleppeid;
- defineerida ettevõttesisesed rolle (arendaja, testija, jt);
- rollidele lisada erinevaid tasemeid (junior, mid, senior, jt);
- kasutajal muuta temaga sõlmitud kokkulepete staatust;
- kasutajal näha enda arenguplaani;
- kasutajal muuta enda profiili;
- arenguestluse läbiviijal määrata, mida teine osapool arenguestluse protokollist näha saab.

4.3.2 Mittefunktsionaalsed nõuded

Erinevalt funktsionaalsetest nõuetest, mis kirjeldavad, kuidas süsteem peab toimima, kirjeldavad mittefunktsionaalsed nõuded kuidas tarkvara midagi tegema peab. Autor on välja toonud 4 mittefunktsionaalset nõuet, mis on olulised loodava rakenduse valmimise juures:

- Kliendirakenduse stiil peab olema läbivalt sarnane.
- Rakendus peab kuvama teateid andmete muutmise ja vigade korral.
- Andmete kustutamisel tuleb kasutajat hoiatada, et andmeid pole võimalik tagasi saada ning saada kinnitus, et kasutaja on kustutamisest teadlik.
- Kliendi ja serveri suhtlus peab käima üle HTTPS protokoll.

4.3.3 Funktsionaalsete nõuete prioritseerimine MoSCoW mudeliga

Kuna nõudeid on palju ja kõikide arendamine võtab palju aega, on oluline nendest eristada, mis pakuvad kliendile rohkem väärtust ning milliste funktsionaalsustega tuleb rakenduse arendamist alustada.

Võttes arvesse küsitlusest saadud kliendi sisendit ning kasutades MoSCoW meetodit [7] ärinõuete prioritseerimiseks, on võimalik nõuded jagada nelja erinevasse gruppi:

- M (must have) – peab olema. Nõuded peavad olema täidetud, et lahendus töötaks.

- S (should have) – peaks olema. Nõuded peaks olema täidetud, kuid lahendus võib toimida ka ilma nendeta.
- C (could have) – võiks olla. Nõuded võiksid olla täidetud, kuid mitte vajalikud lahenduse töötamiseks.
- W (won't have) – ei pea olema. Nõuded ei pea olema täidetud ning prioriteedi nimekirjas on tagapool, mis tihtilugu üldse ei realiseerita.

Funktsionaalsed nõuded on prioriseeritud ja jagatud MoSCoW tabelisse (Tabel 2) aritmeetilise keskmise \bar{x} alusel, mille vahemikud on välja mõeldud autori poolt:

- MUST $\bar{x} \geq 4,5$
- SHOULD $4,0 \leq \bar{x} < 4,5$
- COULD $3,0 \leq \bar{x} < 4,0$
- WON'T $\bar{x} < 3,0$

Tabel 2. Funktsionaalsete nõuete prioritseerimine MoSCoW mudeli abil.

PEAB OLEMA (MUST)	PEAKS OLEMA (SHOULD)
<ul style="list-style-type: none"> - Rakendus peab võimaldama kasutajatele saata teateid ja meeldetuletusi. - Rakendus peab võimaldama töökeele vahetamist inglise keelde. - Rakendus peab võimaldama arenguestluse läbiviijal lisada arenguestluse käigus personaalseid märkmeid. - Rakendus peab võimaldama kasutajal näha arenguestluste käigus sõlmitud kokkuleppeid. 	<ul style="list-style-type: none"> - Rakendus peab võimaldama meeskondade loomist olemasolevatest kasutajatest. - Rakendus peab võimaldama ettevõtte haldaja kontoga õiguste andmist teistele kasutajatele. - Rakendus peab võimaldama töötajatel anda tagasisidet ettevõttele. - Rakendus peab võimaldama töötajatel teha ettepanekuid ettevõttele. - Rakendus peab võimaldama koostada malle arenguestluste läbiviimiseks. - Rakendus peab võimaldama defineerida ettevõttesiseseid rolle (arendaja, testija, jt). - Rakendus peab võimaldama kasutajal näha enda arenguplaani. - Rakendus peab võimaldama kasutajal muuta enda profiili.

VÕIKS OLLA (COULD)	EI PEA OLEMA (WON'T)
<ul style="list-style-type: none"> - Rakendus peab omama töölauda, kus on statistika ettevõtte töötajate kohta. - Rakendus peab võimaldama integratsiooni personalihaldus tarkvaraga. - Rakendus peab võimaldama kasutajate importimist rakendusse. - Rakendus peab olema kasutajasõbralik vaegnägijatele ja kuulmispuudega inimestele. - Rakendus peab võimaldama valida erinevat tüüpi arenguestluste vahel. - Rakendus peab võimaldama eksporti varasemalt läbiviidud arenguestlustele (failiformaatiks json, csv). - Rakendus peab võimaldama rollidele lisada erinevaid tasemeid (junior, mid, senior, jt). 	<ul style="list-style-type: none"> - Rakendus peab võimaldama ettevõtte uudisvoo lisamist. - Rakendus peab võimaldama töökeele vahetamist vene keelde. - Rakendust peab olema võimalik kasutada võrguühenduseta. - Rakendus peab võimaldama sõnumivahetust teiste kasutajatega.

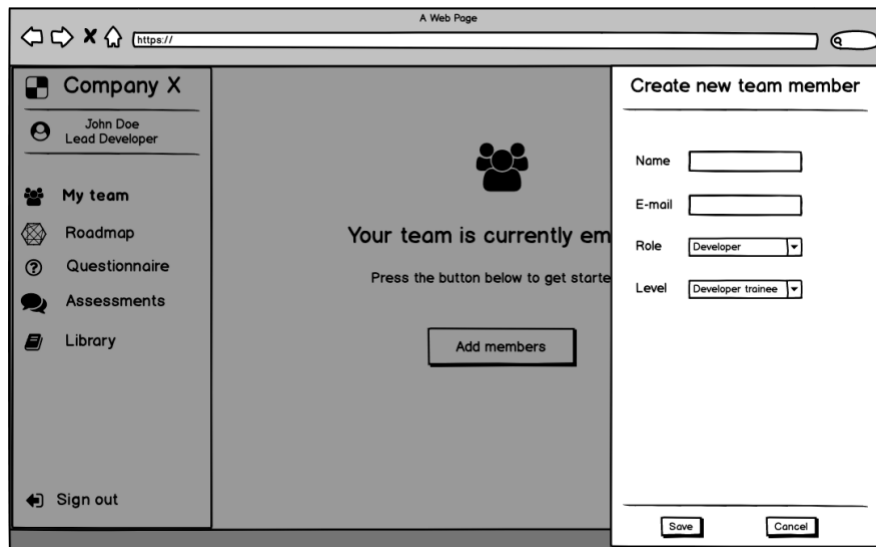
4.3.4 Planeeritava rakenduse prototüüp

Prototüüp on arendatava tarkvara mittetäielik versioon [8]. Protsessi testimiseks loodud esialgne toode, mis võib olla funktsionaalne või näiteks paberi peal visualiseeritud vaade. Eesmärk on luua loodava rakenduse visuaalne näidis. Prototüüpimine on hea vahend tarkvara tellivale kliendile anda edasi omapoolne visuaalne nägemus loodavast tarkvarast.

Käesoleva rakenduse prototüüp on valmistatud kasutades veebipõhist rakendust Balsamiq Cloud [9], mis pakub 30 päevast tasuta prooviperioodi. Oma rohkete valmislahenduste ja ilusate visuaalsete elementidega sobis see suurepäraselt käesoleva rakenduse prototüübi loomiseks.

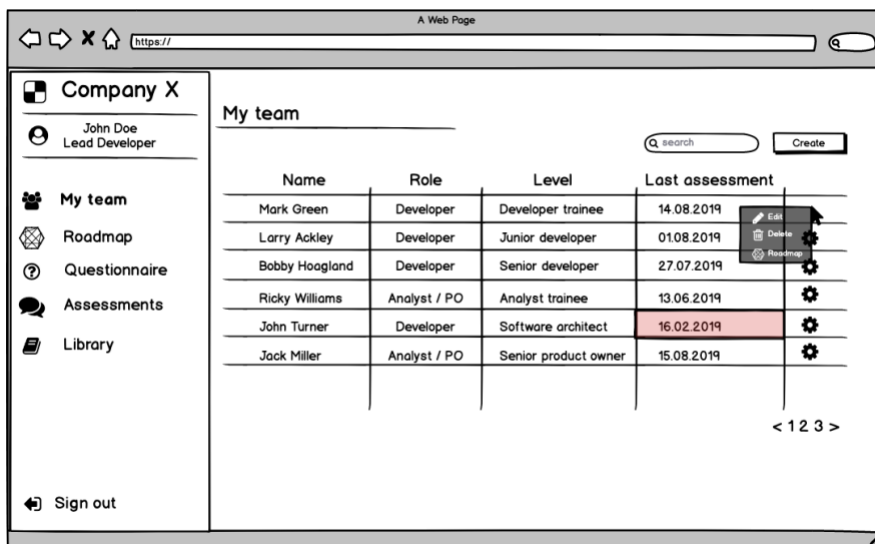
4.3.5 Planeeritava rakenduse prototüübi joonised

Alljärgneval joonisel (Joonis 5) on näha ettevõtte haldaja rollis olevat isik, kes on oma ettevõttesiseses meeskonda lisamas esimest liiget. Kohustuslike veldena on isiku nimi, e-mail, roll ettevõttes ning tema taseme järk selles rollis.



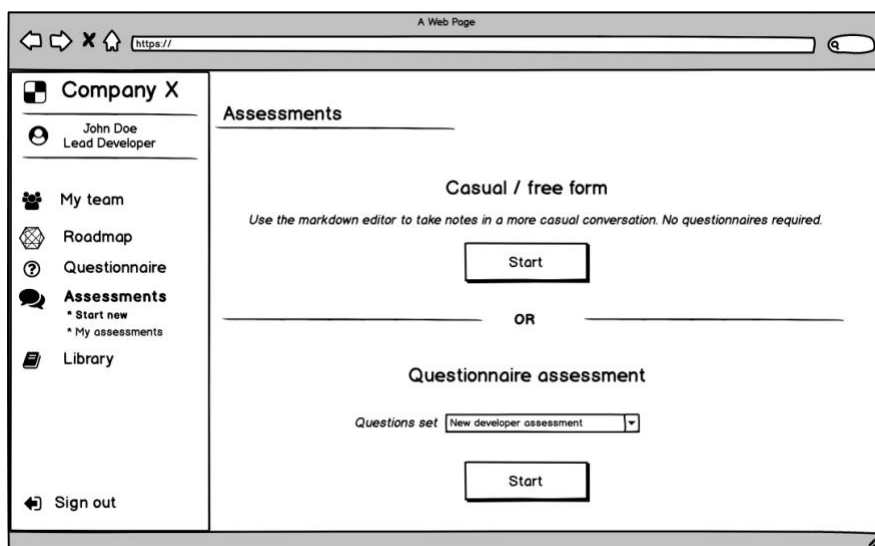
Joonis 5. Prototüübi ekraanivaade uue isiku lisamisel.

Pärast edukat uue meeskonnaliikme lisamist on näha „Minu meeskond“ tabel (Joonis 6), kus on lisaks eelnevale infole välja toodud ka viimase arenguestluse kuupäev. Ettevõtte haldajal jääb võimalus kasutaja infot muuta, kustutada või navigeerida tema arenguplaani vaatesse.



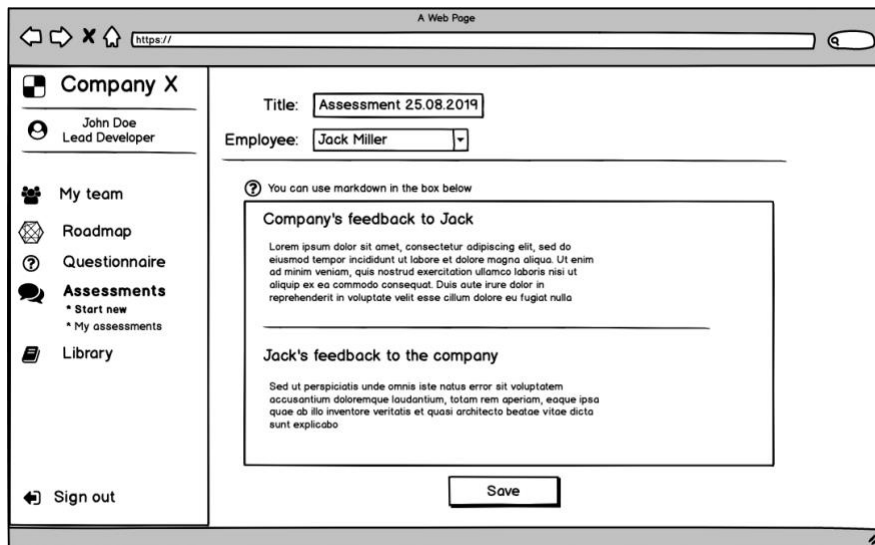
Joonis 6. Prototüübi ekraanivaade meeskonna koosseisust.

Alljärgneval ekraanikuval (Joonis 7) võib näha ettevõtte haldaja poolt antud volitustega isik, kellel on õigus ettevõtte töötajatega läbi viia arenguveestlusi. Võimalik on valida vabas vormis arenguveestluse või eeltäidetud küsimustiku vahel.



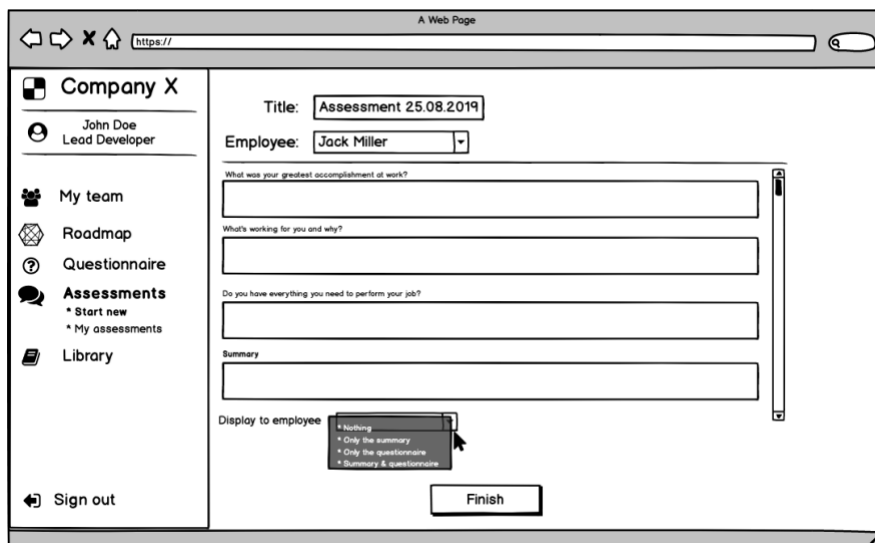
Joonis 7. Prototüübi ekraanivaade arenguveestluse alustamisel.

Alljärgneval joonisel (Joonis 8) on näha, et arenguveestluse läbiviimiseks on valitud vabas vormis vestlus. Läbiviijale avanevas vaates on võimalik määrata arenguveestluse pealkiri, valida töötajate nimekirjast isik ning teha märkusi vaba teksti välja.



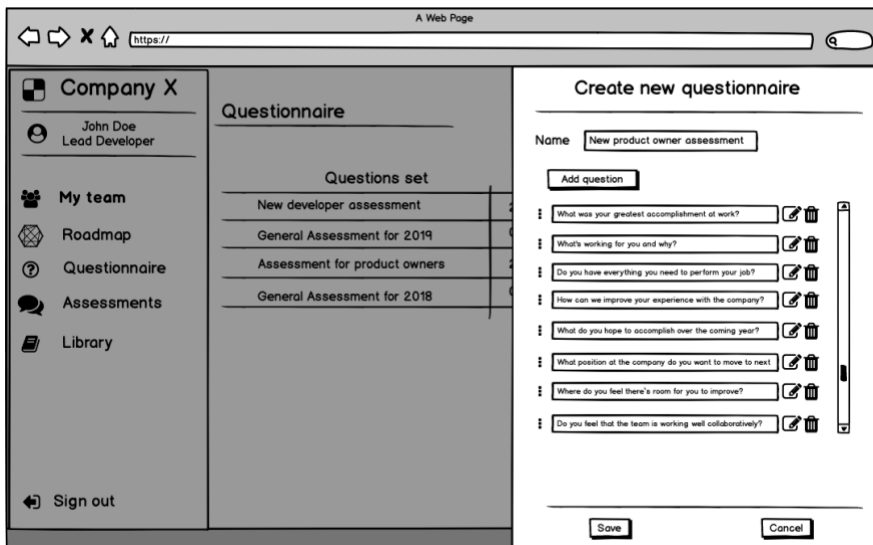
Joonis 8. Prototüübi ekraanivaade vabas vormis arenguestluse läbiviimisel.

Järgneval joonisel (Joonis 9) on aga valitud arenguestluse läbiviimine kasutades eeltäidetud malli. Peamine info on sama, mis vabas vormis arenguestlusel, kuid vaba teksti välja asemel on eeltäidetud küsimustik.



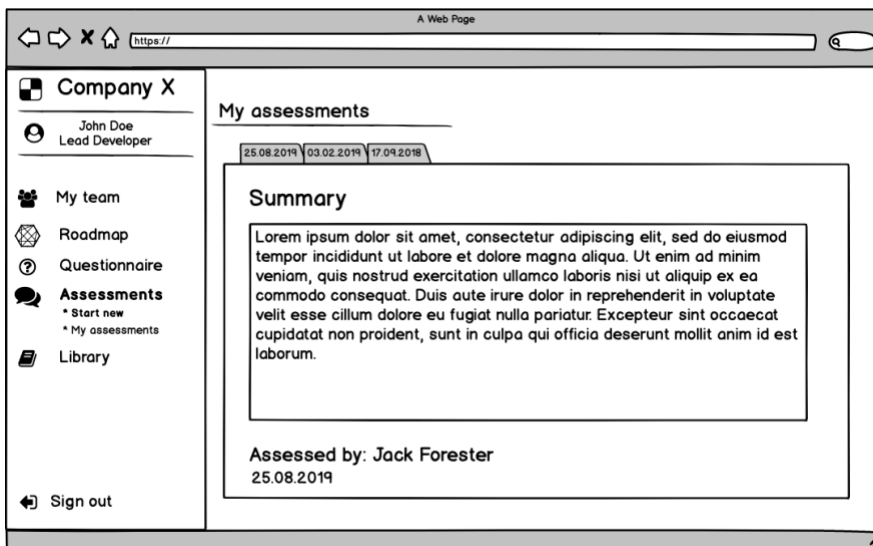
Joonis 9. Prototüübi ekraanivaade arenguestluse läbiviimisel kasutades küsimustiku.

Alljärgneval joonisel (Joonis 10) on kujutatud arenguestluse uue malli loomise vaade, mille taustal on näha varasemalt koostatud mallid. Uute arenguestluste mallide loomise õigus on vaikimisi ettevõtte haldajal või tema poolt volitatud isikul. Kõiki malle on võimalik ka muuta või kustutada.



Joonis 10. Prototüübi ekraanivaade uue küsimustiku koostamisel.

Alljärgnev joonis (Joonis 11) on nähtav kõikidele ettevõtte töötajatele. Selles vaates on võimalik näha varasemalt läbiviidud arenguveestluste ajalugu, kuupäeva millal see toimus, kes selle läbi viis ning kokkuvõtlik tekst arenguveestlusest, mille on kirja pannud arenguveestluste läbiviija.



Joonis 11. Prototüübi ekraanivaade varasemalt läbiviidud arenguveestlustest.

4.3.6 Kasutusmallid

Kasutusmall on toimingute jada tegija ja süsteemi vahel, mis kirjeldab soovitatavat funktsionaalsust. Käesolevas peatükis kirjeldab autor kolme kasutusmalli (*use case*).

Tabel 3 kirjeldab kasutusmalli, kui ettevõtte haldaja rollis olev isik lisab süsteemi uue ettevõtte kasutaja.

Tabel 3. Kasutusmall „Uue isiku lisamine“.

Kasutusmalli nimi	KM 1 – Uue isiku lisamine
Lühikirjeldus	Isiku lisamine ettevõtte andmebaasi
Rollid	Ettevõtte haldaja
Eeltingimused	Ettevõtte haldaja on rakendusse sisseloginud. Ettevõtte haldaja teab lisatava isiku eesnime, perekonnanime ja e-posti aadressi.
Põhivoog	<ol style="list-style-type: none">1. Ettevõtte haldaja valib peamenüüst „Ettevõtte töötajad“2. Vajutab ekraanil olevale nupule „Lisa isik“, mille peale avaneb vorm kasutaja lisamiseks3. Sisestab isiku eesnime4. Sisestab isiku perekonnanime5. Sisestab isiku e-posti aadressi6. Vajutab nuppu „Salvesta“
Järelingimused	Uus isik on lisatud ettevõtte süsteemi. Isik saab e-posti aadressile kirja, kus on informatsioon parooli ja profiili info seadistamise kohta.

Tabel 4 kirjeldab kasutusmalli, kus ettevõtte haldaja rollis olev isik koostab ettevõttesiseseks kasutamiseks uue arengumalli.

Tabel 4. Kasutusmall „Arenguestluse malli koostamine“.

Kasutusmalli nimi	KM 2 – Arenguestluse malli koostamine
Lühikirjeldus	Uue arenguestluse malli lisamine ettevõtte andmebaasi
Rollid	Ettevõtte haldaja
Eeltingimused	Ettevõtte haldaja on rakendusse sisseloginud.
Põhivoog	<ol style="list-style-type: none"> 1. Ettevõtte haldaja valib peamenüüst „Arenguestlused“ 2. Vajutab ekraanil olevale nupule „Lisa arenguestluse mall“, mille peale avaneb uus vaade malli lisamiseks 3. Sisestab arenguestlusele nime 4. Sisestab arenguestlusele lühikirjelduse 5. Vajutab nuppu „Lisa küsimus“ 6. Valib küsimuse tüüpi ja sõnastab küsimuse 7. Kordab punkti 5 ja 6, seniks kuni küsimustik on soovitud kujul valmis 8. Vajutab nuppu „Salvesta“
Järeltingimused	<p>Arenguestluse mall on lisatud ettevõtte süsteemi.</p> <p>Malli on koheselt võimalik kasutada arenguestluse läbiviimise käigus.</p>

Tabel 5 kirjeldab kasutusmalli, kus ettevõtte haldaja rollis olev isik alustab arenguveestlust varasemalt süsteemi lisatud ettevõtte töötajaga.

Tabel 5. Kasutumall „Uue arenguveestluse alustamine“.

Kasutusmalli nimi	KM 3 – Uue arenguveestluse alustamine
Lühikirjeldus	Arenguveestluse läbiviimine ettevõtte töötajaga
Rollid	Ettevõtte haldaja
Eeltingimused	Ettevõtte haldaja on rakendusse sisseloginud. Isik, kellega soovitakse arenguveestlust läbi viia, on lisatud süsteemi. Rakendusse on lisatud mõni arenguveestluse mall.
Põhivoog	<ol style="list-style-type: none"> 1. Ettevõtte haldaja valib peamenüüst „Arenguveestlused“ 2. Vajutab ekraanil olevale nupule „Alusta uut arenguveestlust“ 3. Valib arenguveestluse vormi (Vaba vorm/Küsimustik) 4. Juhul kui on valitud küsimustik, siis valib nimekirjast varasemalt koostatud malli. 5. Valib nimekirjast isiku, kellega arenguveestlust läbi viiakse 6. Sisestab arenguveestlusele kirjelduse 7. Viiakse läbi arenguveestluse protsess, sealjuures vastates küsimustele 8. Täidab lahtri eesmärgid/kokkulepped 9. Määratakse kuupäev järgmiseks arenguveestluseks 10. Vajutab nuppu „Lõpeta arenguveestlus“
Järeltingimused	Läbiviidud arenguveestlus on salvestatud ettevõtte süsteemi. Arenguveestlust läbi viidud isikul on võimalik rakendusse sisselogides näha vestlust, sõlmitud kokkuleppeid, eesmäärke ja järgmise vestluse aega.

5 Kasutatud tehnoloogiad ja raamistikud

Tänapäeval on tehnoloogiate valik veebirakenduste arendamise juures on väga lai ning võimekaid programmeerimiskeeli on mitmeid. Kui sobilik programmeerimiskeel on osutunud valituks, seistakse tihtilugu järgmise keerulise otsuse ees – milline raamistik sobib selle juurde kõige paremini. Projekti hilisemas faasis on programmeerimiskeele või raamistiku välja vahetamine väga mahukas ja aeganõudev töö, mistõttu tuleks paika panna saavutatavad eesmärgid ning võrreldes erinevaid tehnoloogiad teha hoolikalt valik.

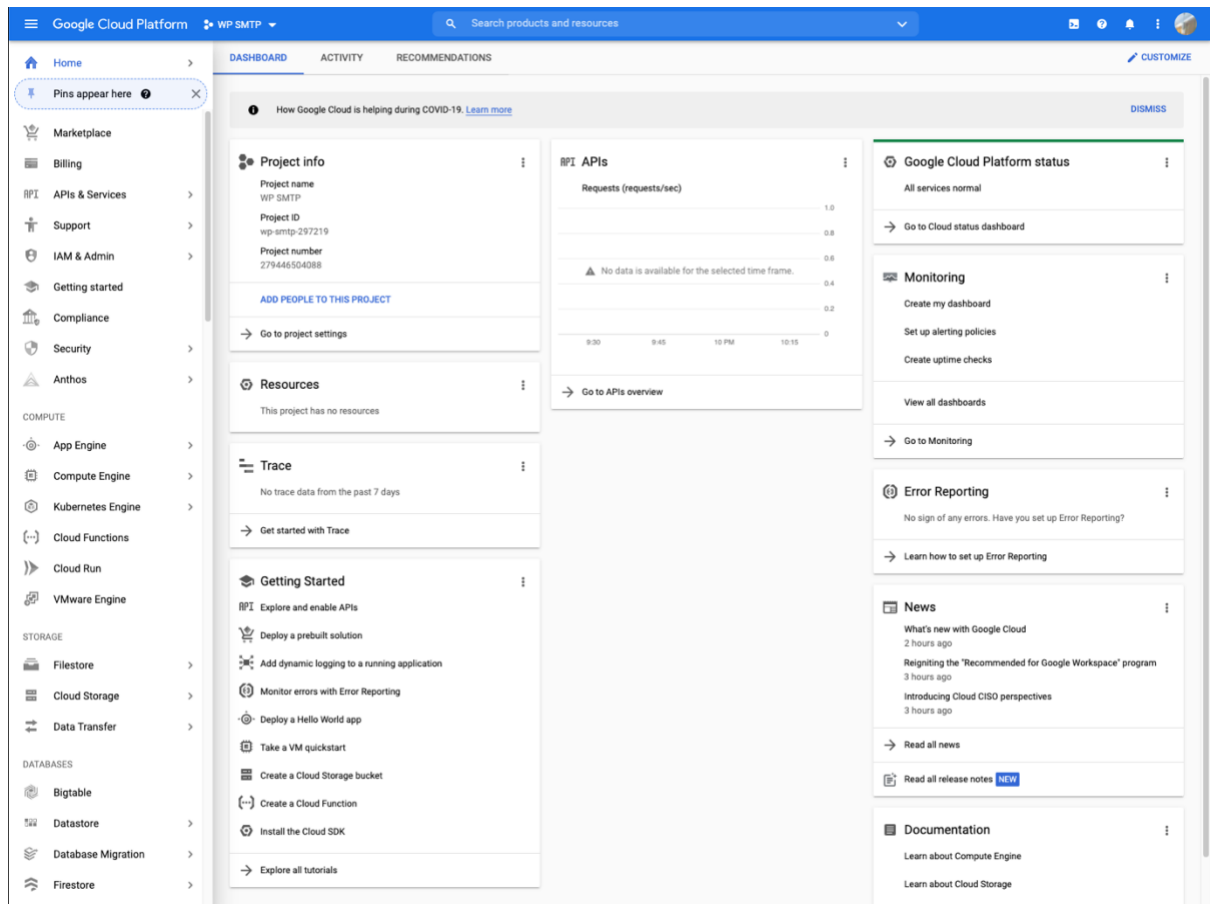
Võttes arvesse piiratud aega käesoleva lõputöö praktilise poole arendamiseks, lähtus autor eelkõige tehnoloogiatest, mis olid varasemalt juba tuttavad.

Käesolevas peatükis kirjeldab autor peamisi tehnoloogiaid ja raamistikke, mis olid kasutusel rakenduse arendamisel.

5.1 Rakenduse majutamine

Rakenduse majutamiseks on kasutatud pilveandmetöötluse platvormi GCP (*Google Cloud Platform*). See on Google poolt välja töötatud aastal 2008 ning omab tänaseks päevaks üle 100 erineva teenuse, millest tuntuimad on majutamiseks *AppEngine*, *Compute Engine* ja serverivaba *Cloud Functions*, failide talletamiseks *Cloud Storage*, andmete hoidmiseks NoSQL andmebaas *Datastore* või suur-andmetega töötlemiseks teenused *BigQuery* või *Pub/Sub*.

Joonis 12 on kuvatõmmis GCP töölauast, kus on näha ka paljud eelnimetatud tooted.



Joonis 12. Google Cloud Platform töölaud.

Käesoleva rakenduse majutamiseks on kasutatud GCP toodet *AppEngine Standard*. See on täielikult ise eskaleeruv toode, mis tähendab, et kui rakenduse koormus järsku kasvab, tehakse automaatselt juurde uusi instantsse, tänu millele ei kannata veebirakenduse kiirus. Kui koormus langeb, siis väheneb ka instantside arv. Kui rakendust ei kasutata, siis aktiivseid instantsse ei ole. See on oluline, kuna teenuse eest maksmine toimub just instantside kasutuse põhised. Et arved liiga suureks ei kasva, on ka oluline seadistada õiged parameetrid serveri-poolsele rakendusele. Rakenduse jaoks on leitud sobilikud parameetrid. (Joonis 13)

```
<instance-class>F1</instance-class>
<automatic-scaling>
  <target-cpu-utilization>0.9</target-cpu-utilization>
  <target-throughput-utilization>0.9</target-throughput-utilization>
  <max-instances>10</max-instances>
  <min-instances>0</min-instances>
  <max-concurrent-requests>10</max-concurrent-requests>
  <max-idle-instances>1</max-idle-instances>
  <max-pending-latency>10000ms</max-pending-latency>
  <min-idle-instances>0</min-idle-instances>
  <min-pending-latency>5000ms</min-pending-latency>
</automatic-scaling>
```

Joonis 13. Rakenduse konfiguratsioon GCP AppEngine jaoks.

AppEngine Standard suurimateks plussideks on lihtne alustamine (uue rakenduse ülesseadmine), suur eskaleeritavus, lihtne evitamine, mille käigus saab kiirelt ja mugavalt suunata liiklust varasemate versioonide peale. Lisaks pakub *AppEngine Standard* automaatset logimist ja rakenduse seiramist. Küll aga on ka mõned puudused. Programmeerimiskeeltest on toetatud on vaid Python, Java, Node.js, PHP, Ruby ja GO [13], mistõttu ei pruugi see sobida kõikidele olemasolevatele projektidele. Nagu paljude teenustega pilvandmetöötlus platvormidele kohane võib ka GCP hinnastamine tunduda esmapilgul keeruline, mis eeldab rohket uurimist ja sobiliku konfiguratsiooni valimist, et vältida üllatavaid ning ebameeldivalt kõrgeid arveid kuu lõpus.

GCP sai valitud eelkõige oma mugavuse, automaatse eskaleerumise ja paindliku hinnapoliitika tõttu.

5.2 Rakenduse serveri-poolne kirjeldus

Käesolevas peatükis kirjeldab autor peamisi serveri-poolseid tehnoloogiaid ja raamistikke, mis olid kasutusel rakenduse arendamisel.

5.2.1 Kotlin

Soome lahes asuva saarestiku järgi nime saanud Kotlin [14] on tasuta, avatud lähtekoodiga, staatiliselt tüübitud programmeerimiskeel, mis kompileerib muuhulgas Java baitkoodiks. Esimene ametlik versioon avaldati 2016 aasta veebruaris [15], kui üks suurimaid arenduskeskkondade loojaid JetBrains [16] ei leidnud piisavalt häid alternatiive oma põhitooto IntelliJ IDEA arendamiseks ning otsustasid kirjutada omale sobilikku. Kotlin võeti kiiresti omaks ning peagi hakati kasutama seda Android rakenduste kirjutamiseks. Kotlin on täielikult ühilduv kõikide olemasolevate Java teekide ja raamistikega, mistõttu peetakse seda ka väga sobilikuks serveri- kui ka töölaua rakenduste arendamiseks. Üheks suureks eeliseks peetakse selle lihtsust ja turvalisust. Samuti soodustab õppimist suurepärase dokumentatsioon [17] ning üha enam kasvav abivalmis kommuun. Kotlin arendati eesmärgiga teha arendajate elu kergemaks ning lahendada ära mõned laialt levinud probleemid, tehes koodi seeläbi ka turvalisemaks ja töökindlamaks – kõrvaldatud on näiteks *null* viidete kasutamine, mida tuntakse ka „The Billion Dollar Mistake“ all [18].

Kotlin sai valitud rakenduse põhiliseks serveri-poolseks programmeerimiskeeleks varasema positiivse kogemuse tõttu ning võrreldes Javaga, saab samaväärse Kotlini koodi kirjutada umbes 40% vähesema ridade arvuga [19].

5.2.2 Spring Boot

Spring Boot on ülekaalukalt kõige populaarsem Java raamistik [20], mille eesmärk on pakkuda arendajale eelseadistatud konfiguratsiooni ning siduda kokku paljud Java raamistikud. Spring Boot on suurepärase abilise seadmaks üles uusi rakendusi – Spring raamistikul põhinev platvorm pakub lisaks eelkomplekteeritud teekide kogumikele ka sisseehitatud rakendusserverit (Tomcat, Jetty või Undertow) [21], mis teeb uue projekti käivitamise arendaja arvutis kiireks ja mugavaks.

Spring Boot raamistiku puuduseks võib välja tuua selle suurt paki mahtu, mille peamiseks põhjustajaks on suur teekide arv ning millest suur osa tihtilugu ei pruugi leida projektis kasutust.

Käesolevas rakenduses on sõltuvustena kasutusel põhiliselt *Spring Security*, *Spring Logging* ja *Spring Boot Starter Web*,

5.2.3 Maven

Apache Maven [22] on projekti ehitamise tööriist, mida kasutatakse eelkõige Java projektides. See põhineb POM-il ning kasutatakse eelkõige projektide arenduskoodi kompileerimiseks ja sõltuvuste süstimiseks. Projekti juurkataloogi tuleb lisada *pom.xml* fail. See XML fail on projekti konfiguratsiooni tuum ning sisaldab endas peamiselt informatsiooni projekti ja konfiguratsiooni kohta, mida läheb tarvis rakenduse ehitamiseks, sealjuures on kirjeldatud rakenduse sõltuvused, ehituskataloog, erinevad pistikprogrammid ning eesmärgid.

```
1. <project>
2.   <modelVersion>4.0.0</modelVersion>
3.
4.   <groupId>com.arenguvestlused.app</groupId>
5.   <artifactId>my-app</artifactId>
6.   <version>1</version>
7. </project>
```

Joonis 14. Minimaalsed nõuded POM jaoks.

Kuna Maven on eelkõige mõeldud kasutamaks Java-le, on oluline täiendada *pom.xml* faili, kohandamaks seda ka Kotliniga, mis on loodava rakenduse põhiline serveri-poolse arenduse programmeerimiskeel. Siinkohal tuleb appi *kotlin-maven-plugin* pistikprogramm, mis kompileerib Kotlini koodi Java baitkoodiks.

Autor kaalus ka alternatiivse ning kiirema projekti ehitamise tööriista Gradle [23] kasutamist, mis on olemuse ja võimaluste poolest ehk isegi pisut parem, kuid võttes arvesse autori varasemat kogemust Maveniga, osutus valituks siiski autorile varasemalt tuttav tehnoloogia.

5.3 Datastore

Google Cloud Datastore on pilvetehnoloogial põhinev automaatselt eskaleeruv NoSQL andmebaasiteenus [24]. Andmete salvestamiseks ja andmebaasipäringute tegemiseks on võimalik kasutada *SQL* süntaksi sarnast *Google Query Language*-t, mis võimaldab teha keerulisi andmebaasipäringuid. Datastore on andmete sisestamisel väga paindlik ning toetab järgmisi andmetüüpe: [25]

- String;
- Date and time;
- Integer;
- Floating point number;
- Boolean;
- Key;
- Geopoint;
- Array;
- Embedded entity;
- Text;
- Null.

Lokaalseks arendamiseks on võimalik paigaldada oma arvutisse *Datastore Emulator*. Nagu ka nimi ütleb, emuleerib see *Google Cloud* toodet *Datastore*, et oleks võimalik mugavalt arendada ja testida olemasolevat koodi, enne kui see *Google Cloud* teenustesse evitada.

Suurimaks *Datastore* puuduseks võiks välja tuua *SQL*-is tuttavat tabelite sidumiseks käsu JOIN puudumist. Siiski on võimalik *üks-mitmele* ja *mitu-mitmele* seose päringuid teha kasutades meetodit *ReferenceProperty*. [26]

Google Cloud Datastore sai valitud projekti andmebaasiks, kuna relatsioonilise andmebaasi, nagu näiteks MySQL või PostgreSQL kasutamine *Google Cloud* keskkonnas oleks olnud antud mahu juures ebamõistlikult kulukas. Lisaks on suureks eeliseks tema kiirus ning vastavalt kasutusele eskaleeritavus.

5.4 Rakenduse kliendi-poolsed tehnoloogiad

Stack Overflow 2020 [27] läbiviidud uuringus, kus osales üle 65 tuhande arendaja üle kogu maailma selgus, et kaheksandat aastat järjest on arendajate seas enim-levinud programmeerimiskeeleks olnud JavaScript [28]. JavaScript on armastatud programmeerimiskeel kliendi-rakenduste loomisel.

Ajalooliselt, kui brauserid ei olnud oma loomu poolest kõige võimsamad, olid JavaScripti kliendirakendused jõudluse poolest üpris kasinad. Iga uus leht küsiti serverist, mispeale laaditi kogu leht uuesti. Tihtilugu võttis see liiga kaua aega ning seetõttu eelistati teisi lahendusi. Tänapäevane modernne JavaScript aga sellist lahendust juba ammu enam ei kasuta. Kliendirakendused on arendatud kui SPA (*Single Page Application*), mille käigus laetakse omale kogu vajaminev HTML, JavaScripti kui ka CSS brauserisse ühe laadimisega. Selle suurimaks eeliseks on kiirus. Enam ei tule oodata millal kliendi-serveri suhtlus jõuab lõpule, et kogu leht uuesti laadida, vaid kogu info küsimine toimub taustal ning kasutaja saab oma toimingutega jätkata kuniks vajaminevad asjad rakendusse laetakse.

ReactJS [29] on JavaScripti teek, mis on mõeldud kasutajaliideste ehitamiseks. Aastal 2013 Facebooki poolt loodud ja välja antud teek on valitud alates aastast 2016 populaarseimaks valikuks kasutajaliideste ehitamiseks, seljatades oma põhilised konkurendid VueJs ja Angular. Omades teadmisi JavaScriptist, on Reacti õppimine üpris lihtne, kuna see on oma olemuselt tavaline JavaScript, koos mõningate lisadega. Taaskasutatavatest komponentidest koosnev ReactJS, võtab sisendi *props* parameetri näol ning tagastab XML süntaksi sarnase struktuuri, mida kutsutakse JSX-ks. Komponentid on võimelised hoidma endas olekuid, mille muutumisel taaslaetakse DOM.

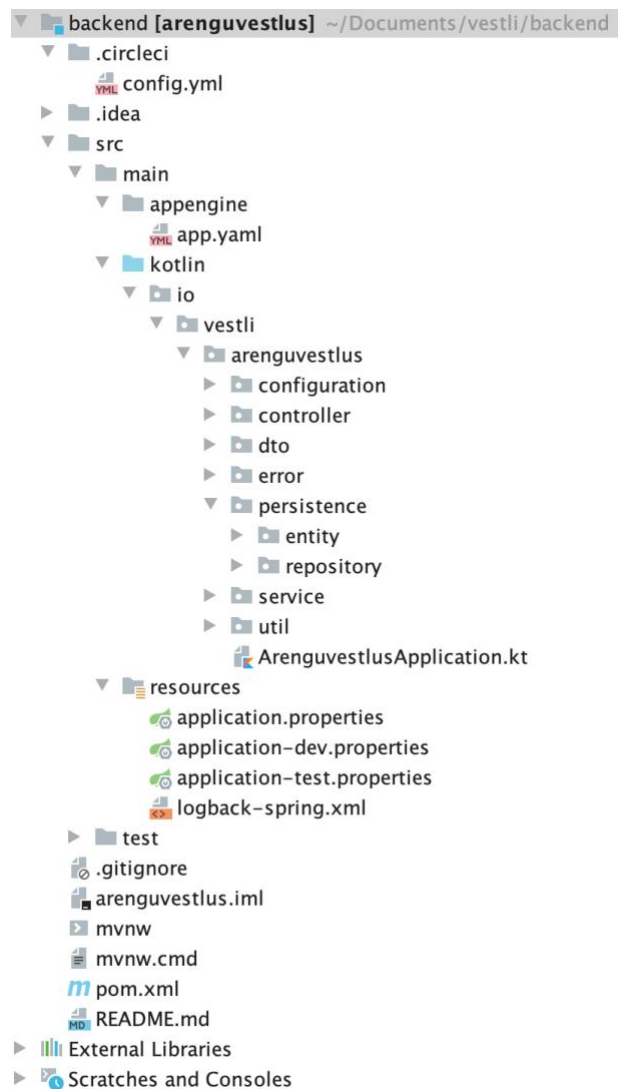
Käesolev kliendirakendus on ehitatud kui *Single Page Application*, kasutades JavaScripti teeki ReactJS.

6 Rakenduse realisatsioon

Käesolevas peatükis tutvustab autor valminud rakenduse ülesehitust ning kirjeldab funktsionaalsuse põhiprotsesse. Lisaks on välja toodud tuleviku arendustegevused.

6.1 Serverirakendus

Serveri-poolne rakendus koosneb kaheksast alamkaustast ning käesolevas peatükis teeb autor põgusa ülevaate, millest need koosnevad. Serveri-poolse projekti struktuuri on näha alljärgneval joonisel (Joonis 15).



Joonis 15. Serveri-poolne projekti struktuur.

configuration on kaust, kuhu kuuluvad eelkõige Springi poolsed konfiguratsiooni failid. Seal toimub näiteks Spring Security konfigureerimine, mille käigus toimub kasutajate autentimine ning kontroll, et lõpp-punktid oleksid kättesaadavad ainult autoriseeritud kasutajatele.

controller kaust sisaldab endas kontrolleri klasse, mis võimaldab suhtlust kliendirakendusega. Kontrolleri kaustas olevad klassid on tähistatud annotatsiooniga `@RestController`. Selles klassis võib olla mitmeid meetodit, mille poole on võimalik pöörduda üle HTTP päringute. Selleks on meetoditele lisatud annotatsioon vastavalt HTTP tüübile. Alljärgneval joonisel (Joonis 16) on HTTP GET tüüpi meetod annotatsiooniga `@GetMapping`, mis võtab parameetriks meetodi marsruudi.

```
package io.vestli.arenguestlus.controller

import io.vestli.arenguestlus.service.UserService
import org.springframework.http.ResponseEntity
import org.springframework.web.bind.annotation.GetMapping
import org.springframework.web.bind.annotation.RestController

@RestController
class UserController (
    private val userService: UserService
) {

    @GetMapping( ...value: "/user/profile")
    fun getUserProfile(): ResponseEntity<Any> {
        val userProfile :UserProfile = userService.getUserProfile()
        return ResponseEntity.ok(userProfile)
    }
}
```

Joonis 16. Kontrolleri klassi meetod kasutaja profiili küsimiseks.

dto kaust sisaldab endas andmete edastamise objekte (*Data Transfer Object*) [30]. Seda laadi objektide ülesanne on kanda andmeid erinevate protsesside vahel.

error kaust sisaldab endas veatöötuse halduseks mõeldud klasse.

persistence koosneb kahest alamkaustast:

- **entity** kaust sisaldab endas andmebaasiga suhtlemiseks domeeni objekte.
- **repository** kaust sisaldab endas andmepääsu kihi loogikat.

service kaust sisaldab endas rakenduse põhilist ärioloogikat. Klassid selles kaustas on üldjuhul tähistatud Spring annotatsiooniga @Service.

util kaust sisaldab endas ühiskasutatavaid abiklasse.

6.2 Rakenduse API lõpp-punktid

Rakendus koosneb erinevatest kontrollieritest serveri-poolses rakenduses, kuhu poole on võimalik pöörduda varasemalt autoriseeritud kasutajal *Bearer* tokeni abil.

Välja ei ole toodud kõiki API lõpp-punkte, kuna paljude objektide vaatamine, loomine, muutmine ja kustutamine koosneb sarnasest ülesehitusest, mille juures on vaid objekt erinev.

Tabel 6. Rakenduse lõpp-punktide loetelu.

Lõpp-punkt	Päringu HTTP meetod	Kirjeldus
/signin	POST	Logib kasutajanime ja parooli alusel sisse ning tagastab JWT koos sisseloginud kasutaja üldinfoga.
/users	GET	Tagastab loetelu ettevõtte kasutajatest.
/user/profile/{id}	GET	Tagastab ettevõtte kasutaja info id alusel.
/user/profile	POST	Uue kasutaja loomine
/user/profile/{id}	PUT	Olemasoleva kasutaja muutmine id alusel.
/user/profile/{id}	DELETE	Olemasoleva kasutaja kustutamine id alusel.
/teams	GET	Tagastab loetelu ettevõtte meeskondadest.
/team	POST	Uue meeskonna loomine.
/assessments	GET	Tagastab loetelu ettevõtte arenguestluste mallidest.
/assessment/template	POST	Uue arenguestluse malli loomine
/assessment/template/{id}	DELETE	Arengumalli kustutamine id alusel.
/user/{id}/assessments	GET	Tagastab loetelu kasutaja arenguestlustest id alusel.
/user/{id}/assessment	POST	Läbiviidud arenguestluse info salvestamine kasutaja id alusel.

6.3 Rakenduse kasutajate rollid

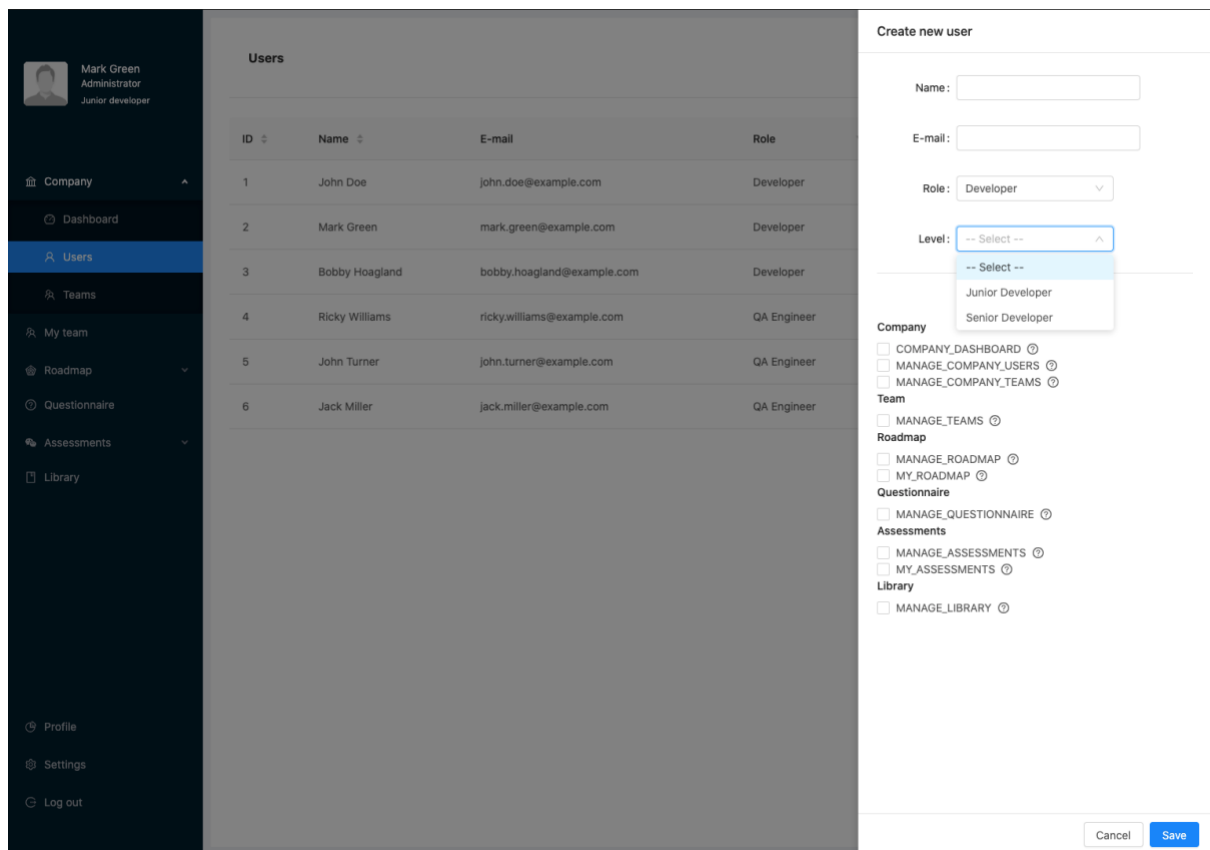
Rakenduse haldaja (ADMIN) on kõrgeima õigustega roll rakenduses. Rakenduse haldaja saab lisada süsteemi uusi ettevõtteid ning määrata ettevõttele vastutava isiku, kasutades selleks vaid temale mõeldud vaadet kliendirakenduses ning turvatud lõpp-punkti serverirakenduses. Rakenduse haldaja rollis oleval isikul on võimalik näha süsteemiga liitunud ettevõtteid.

Ettevõtte haldaja (COMPANY_MANAGER) rollis oleval isikul on kõik õigused teha ettevõttega seotud toiminguid. Ettevõtte haldajal on võimalus lisada süsteemi uusi kasutajaid ning määrata neile süsteemis õigusi, mis võimaldab ligipääsu erinevatele ressurssidele ja toimingutele. Ettevõtte haldaja saab koostada meeskondi, luua arenguvestluse malle ning läbi viia arenguvestlusi.

Tavakasutaja (USER) õigused süsteemis on rohkem piiratud. Tavakasutaja rollis oleval isik saab muuta oma profiili infot ning kontaktandmeid. Peamenüüs avaneb tavakasutajale menüüpunkt „Minu arenguvestlused“, millele vajutades navigeeritakse kasutaja uuele vaatele, kus on võimalik tutvuda varasemalt läbi viidud arenguvestlustega. Valides kaardilt konkreetse arenguvestluse, avaneb kasutajale ülevaade, millal arenguvestlus leidis aset, kes oli selle läbiviijaks, näha märkmeid ning vestluse käigus sõlmitud kokkuleppeid.

6.4 Kasutaja lisamine, meeskonna loomine ja õiguste jagamine

Ettevõtte haldaja rollis oleval isikul on võimalik lisada ettevõtte süsteemi uusi kasutajaid. Selleks tuleb navigeerida peamenüüst „Kasutajad“ menüüpunkti ning vajutada „Lisa kasutaja“ nuppu juba süsteemi sisestatud kasutajate tabeli kõrval. Kui ettevõtte süsteemi pole ühtegi kasutajat varasemalt lisatud, siis tühja tabelit ei kuvata. Lisades uut kasutajat on kohustuslike väljadena tarvis lisada isiku nimi, e-mail, roll ettevõttes ning tase selles rollis. Lisaks on võimalik määrata kasutajale ka õigused süsteemis, millistele ressurssidele ta rakenduses ligi pääseb. (Joonis 17)



Joonis 17. Kuvatõmmis rakendusest uue kasutaja lisamisel.

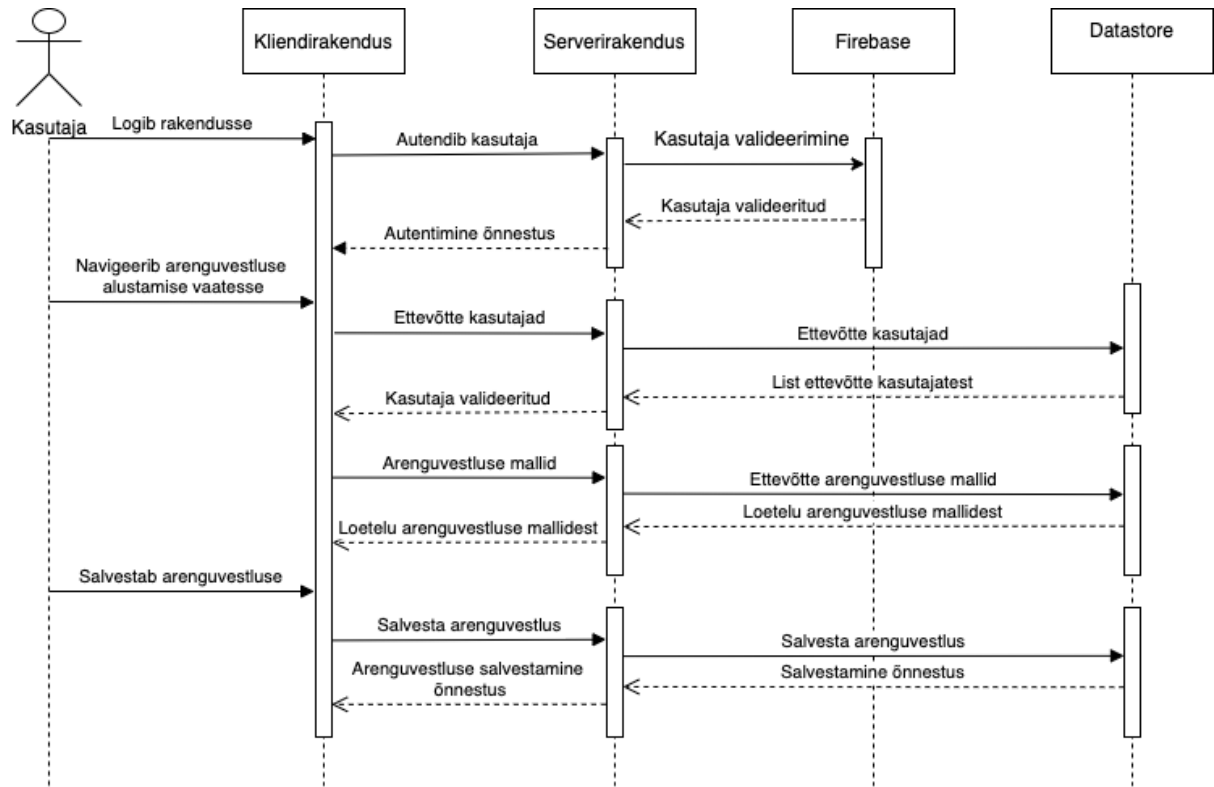
Ettevõtte süsteemi lisatud kasutajatest on võimalik ettevõtte haldaja rollis oleval isikul luua meeskondi. Meeskondade või selle liikmete arv ei ole piiratud ning meeskonnal peab olema juht. Juhiks võib olla näiteks väiksema ettevõtte puhul haldaja ise või suurema ettevõtte puhul tiimijuht. Ettevõtte koosseisu on võimalik igal hetkel muuta ning kui ettevõtetest lahkub töötaja, eemaldatakse ta ka meeskonna koosseisust.

6.5 Arenguestluste läbiviimine

Arenguestluste läbiviimise protsess algab kasutaja sisselogimisega. Kui kasutajale on antud süsteemis õigused läbi viimaks arenguestluste, on peamenüüs nähtav menüüpunkt „Arenguestlused“, mille alam-menüüpunkt on „Alusta arenguestlustega“. Seejärel on võimalus valid kahte tüüpi arenguestluste vahel – kas vabas vormis arenguestlustus või kasutades varasemalt loodud arenguestluste malli. Tehes oma valiku, tuleb järgmises vaates sisestada arenguestluste pealkiri või kirjeldus ning valida ettevõtte töötajate nimekirjast inimene, kellega arenguestlust läbi viiakse. Kui varasemalt valiti vabas vormis arenguestlustus, siis avaneb arenguestluste läbiviijal tekstivorm, kuhu saab märkmeid teha. Kui valiti mõni

varasemalt loodud mall, tuleb tekstivormi asemel küsimustik. Arenguevestluse lõppedes tuleb vajutada lehekülje all asuvat nuppu „Salvesta“, mis talletab arenguevestluse info andmebaasi.

Joonisel 18 on näha arenguevestluste läbiviimise jadaskeem.



Joonis 18. Arenguevestluste läbiviimise jadaskeem.

7 Tuleviku arendustegevused

Lõputöö käigus sai töö autor ülevaate, millised on Eesti IT-ettevõtete ootused arenguvestluste läbiviimist toetava tarkvara loomisel. Prioriseeritud ärinõuetest sai enamus kõrgeima prioriteediga funktsionaalsused implementeeritud ka rakendusse. Kuna tervikliku veebirakenduse arendamine on väga mahukas töö ning olenevalt projektist võib nõuda tuhandeid arendustunde [31], siis ka käesoleva lõputöö raames valmis vaid osa planeeritust.

Kui käesoleva töö käigus keskendus töö autor eelkõige server-poolse rakenduse arendamisele, siis tulevikuks on planeeritud juba olemasoleva kasutajaliidese mugavamaks ning rohkem kasutajasõbralikumaks tegemine. Plaanis on luua ülevaatlik töölaud ettevõtte haldajale, kus oleksid nähtavad erinevad graafikud ja skeemid, sealjuures kui palju on näiteks viimase kvartali või mõne muu valitud perioodi jooksul vestlusi juba läbi viidud ning mitu arenguvestlust veel on ees. Kindlasti tuleks keskenduda ka rollide jaotusele ning võimaldada arenguvestlusi läbi viia isikutel, ilma et nad näeksid muud olulist infot ettevõtte kohta. Kui tavakasutaja võimalused rakenduses on hetkel üpris piiratud, siis plaanis on luua põhjalik enesearengu plaan töötajale, kus oleks võimalik seada endale ise eesmärged ning kaardistada arengu protsessi. Viimase suure funktsionaalsusena toob töö autor välja ka meeldetuletuste võimaluse, mis teeks kokkuvõtte viimasest arenguvestlusest ning kokkuleppetest ja annaks märku kui uus arenguvestlus on lähenemas.

8 Kokkuvõte

Bakalaureusetöö eesmärgiks oli kaardistada Eesti IT-ettevõtete ootused arenguestlust toetava tarkvara rakendusele ning lähtuvalt uuringust saadud vastuste põhjal luua toimiv rakendus, mis vastab Eesti IT-ettevõtete vajadustele ja ootustele.

Lõputöö eesmärgi saavutamiseks koostas autor veebipõhise küsitluse, mille käigus uuris, kas Eesti IT-ettevõtted viivad läbi arenguestlusti oma organisatsioonis ning millisel kujul nad seda teevad. Lisaks uuriti, millised on ettevõtete ootuseid loodava tarkvara kohta. Saadud tulemused prioritseeriti ning selgitati välja peamised funktsionaalsed ja mittefunktsionaalsed nõuded. Valmis rakenduse prototüüp ning töö autor kirjeldas loodava rakenduse valmimisel kasutatud tehnoloogiaid. Töö praktilise osana valmis arenguestluste läbiviimist toetav tarkvara.

Lõputöö käigus on valminud kliendi- kui ka serveri-poolne rakendus. Võimalik on lisada rakendusse uusi ettevõtteid ning sellele vastutav isik. Ettevõtte haldaja rollis olev isik saab lisada süsteemi uusi inimesi ning koostada nendest meeskondi. Lisaks vabast vormis arenguestlustele, on võimalik koostada erinevaid malle, mida saab kasutada arenguestluste läbiviimiseks. Kasutajal on võimalik näha varasemalt läbiviidud arenguestluste ajalugu, kokkuleppeid ning seatud eesmärke. Käesoleva lõputöö raames on valminud rakenduse esimene versioon ning töö autoril on plaan olemasolevat rakendust edasi arendada. Näiteks on planeeritud olemasoleva kasutajaliidese mugavamaks ning rohkem kasutajasõbralikumaks tegemine. Lisaks ka meeldetuletuste võimalus, mis teeks kokkuvõtte viimasest arenguestlustest ning kokkuleppetest ja annaks märku kui uus arenguestlust on lähenemas.

Autori hinnangul said bakalaureusetöö eesmärgid täidetud. Küsitluse käigus sai ettevõtetelt kogutud ka kontaktinfot, kuhu poole pöörduda, kui rakendus on valmis kasutamiseks, et seda ettevõtetele tutvustada.

Kasutatud kirjandus

- [1] „Arenguvestluse korraldus - Arenguvestlused ettevõttes,“ [Võrgumaterjal]. Available: <https://arenguvestlus.weebly.com/arenguvestluse-korraldus.html> [Kasutatud Aprill 2021].
- [2] „Inimressursi juhtimine,“ [Võrgumaterjal]. Available: https://www.researchgate.net/publication/48341997_Inimressursi_juhtimine. [Kasutatud Aprill 2021].
- [3] Stone, R. 1998. Human Resource Management. 3rd ed. John Wiley & Sons.
- [4] Fisher, C., Schoenfeldt, L., Shaw, J. 1999. Human Resource Management. 4th ed. Houghton Mifflin Company.
- [5] „Ametialaste arenguplaanide digitaliseerimine Helmeses,“ [Võrgumaterjal]. Available: <https://dspace.ut.ee/handle/10062/56242>. [Kasutatud Aprill 2021].
- [6] „Arenguvestluste efektiivsuse hindamise kriteeriumid ja kasuteguri suurendamise võimalused koolis,“ [Võrgumaterjal]. Available: <https://dspace.ut.ee/handle/10062/17922>. [Kasutatud Aprill 2021].
- [7] „What is MoSCoW Prioritization? | Overview of the MoSCoW Method,“ [Võrgumaterjal]. Available: <https://www.productplan.com/glossary/moscow-prioritization/>. [Kasutatud Aprill 2021].
- [8] C. Lauff, M. Rentschler ja D. Kotys-Schwartz, „What is a Prototype? What are the Roles of Prototypes in Companies?,“ Journal of Mechanical Design, kd. 140, lk. 1-12, 2018.
- [9] „Balsamiq. Rapid, Effective and Fun Wireframing Software | Balsamiq,“ [Võrgumaterjal]. Available: <https://balsamiq.com/>. [Kasutatud Aprill 2021].
- [10] „Planeerimise, projektijuhtimise ja arenguvestluste tarkvara | PlanPro,“ [Võrgumaterjal]. Available: <https://planpro.ee/>. [Kasutatud Aprill 2021].
- [11] „Talendijuhtimine ja tagasiside | Upsteem.com,“ [Võrgumaterjal]. Available: <https://www.upsteem.com/>. [Kasutatud Aprill 2021].

- [12] „Google Sheets: Free Online Spreadsheets for Personal Use,“ [Võrgumaterjal]. Available: <https://www.google.com/sheets/about/>. [Kasutatud Aprill 2021].
- [13] “Choosing an App Engine environment, ” [Võrgumaterjal]. Available: <https://cloud.google.com/appengine/docs/the-appengine-environments>. [Kasutatud Aprill 2021].
- [14] „Kotlin Programming Language,“ [Võrgumaterjal]. Available: <https://kotlinlang.org/>. [Kasutatud Aprill 2021].
- [15] „Kotlin 1.0 Released: Pragmatic Language for JVM and Android,“ [Võrgumaterjal]. Available: <https://blog.jetbrains.com/kotlin/2016/02/kotlin-1-0-released-pragmatic-language-for-jvm-and-android/>. [Kasutatud Aprill 2021].
- [16] „Jetbrains corporate overview,“ [Võrgumaterjal]. Available: https://resources.jetbrains.com/storage/products/jetbrains/docs/corporate-overview/en-us/jetbrains_corporate_overview.pdf. [Kasutatud Aprill 2021].
- [17] „Get started with Kotlin | Kotlin,“ [Võrgumaterjal]. Available: <https://kotlinlang.org/docs/getting-started.html>. [Kasutatud Aprill 2021].
- [18] „Null References: The Billion Dollar Mistake,“ [Võrgumaterjal]. Available: <https://www.infoq.com/presentations/Null-References-The-Billion-Dollar-Mistake-Tony-Hoare/>. [Kasutatud Aprill 2021].
- [19] „What advantages does Kotlin give me over the Java programming language?,“ [Võrgumaterjal]. Available: <https://kotlinlang.org/docs/faq.html#what-advantages-does-kotlin-give-me-over-the-java-programming-language>. [Kasutatud Aprill 2021].
- [20] „Top 4 Java Web Frameworks Revealed: Real Life Usage Data of Spring MVC, Vaadin, GWT and JSF,“ [Võrgumaterjal]. Available: <https://www.jrebel.com/blog/java-web-framework-usage-stats>. [Kasutatud Aprill 2021].
- [21] „Spring Boot,“ [Võrgumaterjal]. Available: <https://spring.io/projects/spring-boot>. [Kasutatud Aprill 2021].
- [22] „Maven; Welcome to Apache Maven,“ [Võrgumaterjal]. Available: <https://maven.apache.org/>. [Kasutatud Aprill 2021].

[23] „Gradle Build Tool,“ [Võrgumaterjal]. Available: <https://gradle.org/>. [Kasutatud Aprill 2021].

[24] „Datastore Overview | Cloud Datastore Documentation | Google Cloud,“ [Võrgumaterjal]. Available: <https://cloud.google.com/datastore/docs/concepts/overview>. [Kasutatud Aprill 2021].

[25] „Entities, Properties, and Keys | Cloud Datastore Documentation,“ [Võrgumaterjal]. Available: <https://cloud.google.com/datastore/docs/concepts/entities>. [Kasutatud Aprill 2021].

[26] „Modeling entity relationships on App Engine standard environment,“ [Võrgumaterjal]. Available: <https://cloud.google.com/community/tutorials/appengine-modeling-entity-relationships>. [Kasutatud Aprill 2021].

[27] „Stack Overflow Survey 2020 - Programming, Scripting, and Markup Languages,“ [Võrgumaterjal]. Available: <https://insights.stackoverflow.com/survey/2020#technology-programming-scripting-and-markup-languages-all-respondents>. [Kasutatud Aprill 2021].

[28] „JavaScript.com,“ [Võrgumaterjal]. Available: <https://www.javascript.com/>. [Kasutatud Aprill 2021].

[29] „React – A JavaScript library for building user interfaces,“ [Võrgumaterjal]. Available: <https://reactjs.org/>. [Kasutatud Aprill 2021].

[30] „Data Transfer Object | Microsoft Docs,“ [Võrgumaterjal]. Available: [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649585\(v=pandp.10\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649585(v=pandp.10)?redirectedfrom=MSDN). [Kasutatud Aprill 2021].

[31] „How Long Does It Take to Develop a Web App: Answers to All Questions,“ [Võrgumaterjal]. Available: <https://gbksoft.com/blog/how-long-does-it-take-to-develop-a-web-app/>. [Kasutatud Aprill 2021].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Renar Tupits

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Arenguvestluste läbiviimist toetava tarkvara loomine Eesti IT-ettevõtete näitel", mille juhendaja on Kristiina Hakk
 - 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.

Lisa 2 – Küsimustik IT-ettevõtetele

Tere!

Olen Tallinna Tehnikaülikooli IT-süsteemide arenduse üliõpilane Renar Tupits ning kutsun Teid osalema uuringus, mille eesmärk on kaardistada Eesti IT-ettevõtete ootused arenguestluste läbiviimist toetava tarkvara loomiseks. Uuring on üks osa minu bakalaaurusetööst ning selle tulemusena valmib arenguestluste läbiviimist toetava tarkvara rakendus. Küsimustiku täitmine võtab aega umbes viis minutit. Uuringus osalemine on anonüümne ning saadud vastuseid esitatakse vaid üldistatud kujul.

Täiendavate küsimuste korral võtke palun ühendust.

Renar Tupits

Tallinna Tehnikaülikooli IT-süsteemide arenduse III kursuse üliõpilane

renartupits@gmail.com

1. Mitu töötajat on Teie ettevõttes?
 - 1 - 9
 - 10 - 49
 - 50 - 250
 - 250 ja enam

2. Mitu aastat on Teie ettevõtte tegutsenud?
 - Kuni aasta
 - 1 – 5 aastat
 - 6 - 10 aastat
 - 10 aastat ja enam

3. Kas Teie ettevõtte peab oluliseks töötajate arengu toetamist?
 - Jah
 - Ei

4. Kas Teie ettevõttes viiakse töötajatega läbi arenguveestlusi?

- Jah
- Ei

Juhul, kui 4. vastus on Jah.

5. Kas Teie ettevõttes on olemas kindel kord või struktuur arenguveestluste läbiviimiseks?

- Jah
- Ei
- Ei oska öelda
- Other

6. Kes Teie ettevõttes tavapäraselt arenguveestlusi läbi viib?

- Personalijuht
- Tiimijuht
- SCRUM Master
- Vanemarendaja
- Other

7. Millise regulaarsusega viiakse läbi arenguveestlusi Teie ettevõttes?

- Iga kuu
- Iga kvartal
- Iga aasta
- Other

8. Kas ja kuidas hinnatakse arenguveestluste tõhusust/efektiivsust Teie ettevõttes?

9. Kas ja kuidas toimub arenguveestlustel käsitletud teemade ja seatud eesmärkide salvestamine?

- Salvestamine toimub mõnes pilveteenuses
- Salvestamine toimub füüsilisel kuju
- Salvestamine toimub arvutis
- Salvestamist ei toimu
- Other

10. Kas teie ettevõtte kasutab arenguvestluste läbiviimiseks mingit tarkvara?

11. Mis on kasutatava tarkvara nimi?

12. Kas olete selle tarkvaraga rahul?

- Jah
- Ei
- Üldiselt rahul, kuid on ka mõned puudused

13. Kui rahul olete arenguvestluste läbiviimise süsteemiga/korraga oma ettevõttes?

1 - Ei ole üldse rahul

5 - Olen väga Rahul

Juhul, kui 4. vastus on Ei.

14. Mis põhjusel Teie ettevõttes ei viida läbi arenguvestlusi?

- Ressursi puudus
- Puudub teadmine, kuidas efektiivselt arenguestlusi läbi viia
- Puudub sobilik tarkvara, mis toetaks arenguvestluste läbiviimist
- Other

Funktsionaalsed nõuded loodavale rakendusele

15. Järgnevalt palun Teil hinnata kuivõrd nõustute allpool loetletud väidetega. Iga väide kirjeldab ühte võimaliku rakenduse funktsionaalsust.

- Ei nõustu üldse
- Ei nõustu
- Nii ja naa
- Nõustun
- Nõustun täielikult

- Rakendus peab omama töölauda, kus on statistika ettevõtte töötajate kohta.
- Rakendus peab võimaldama integratsiooni personalihaldus tarkvaraga.

- Rakendus peab võimaldama kasutajate importimist rakendusse.
- Rakendus peab võimaldama meeskondade loomist olemasolevatest kasutajatest.
- Rakendus peab võimaldama ettevõtte haldaja kontoga õiguste andmist teistele kasutajatele.
- Rakendus peab võimaldama kasutajatele saata teateid ja meeldetuletusi.
- Rakendus peab võimaldama töötajatel anda tagasisidet ettevõttele.
- Rakendus peab võimaldama töötajatel teha ettepanekuid ettevõttele.
- Rakendus peab võimaldama ettevõtte uudisvoo lisamist.
- Rakendus peab võimaldama töökeele vahetamist inglise keelde.
- Rakendus peab võimaldama töökeele vahetamist vene keelde.
- Rakendus peab olema kasutajasõbralik vaegnägijatele ja kuulmispuudega inimestele.
- Rakendus peab võimaldama sõnumivahetust teiste kasutajatega.
- Rakendus peab võimaldama valida erinevat tüüpi arenguveestluste vahel.
- Rakendus peab võimaldama koostada malle arenguveestluste läbiviimiseks.
- Rakendus peab võimaldama arenguveestluse läbiviijal lisada arenguveestluse käigus personaalseid märkmeid.
- Rakendus peab võimaldama ekspordi varasemalt läbiviidud arenguveestlustele (failiformaatiks json, csv).
- Rakendus peab võimaldama kasutajal näha arenguveestluste käigus sõlmitud kokkuleppeid.
- Rakendus peab võimaldama defineerida ettevõttesiseseid rolle (arendaja, testija, jt).
- Rakendus peab võimaldama rollidele lisada erinevaid tasemeid (junior, mid, senior, jt).
- Rakendus peab võimaldama kasutajal muuta temaga sõlmitud kokkulepete staatust.
- Rakendus peab võimaldama kasutajal näha enda arenguplaani.
- Rakendus peab võimaldama kasutajal muuta enda profiili.
- Rakendus peab võimaldama arenguveestluse läbiviijal määrata, mida teine osapool arenguveestluse protokollist näha saab.

16. Kas Teie hinnangul võiks kirjeldatud omadustega tarkvara leida kasutust Teie ettevõttes?

- Jah

- Ei

17. Kui palju oleks Teie ettevõtte nõus iga kuu maksma kasutamaks sobivat arenguestluste läbiviimise tarkvara?

- 1 – 4 €/kasutaja
- 5 – 9 €/kasutaja
- 10+ €/kasutaja
- Other

18. Kui soovite, et rakenduse valmides võetaks Teiega ühendust ning tehakse demo, siis palun lisage oma kontaktinfo.

