

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Iiris Takel 179669IADB

**AZURE AD'GA INTEGREERITUD  
KESKKONNA LOOMINE TÖÖTAJATE  
KONTAKTIDE HALDAMISEKS  
SHAREPOINTIS**

bakalaureusetöö

Juhendajad: Toomas Lepikult  
Doktor,  
Raimo Seero  
Bakalaureus

Tallinn 2020

## **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: Iris Takel

17.05.2020

## Annotatsioon

Lõputöö eesmärk on luua ettevõttele, kus kõigil töötajatel on Azure AD konto, töötajate kontaktide haldamise keskkond SharePointis. Töö käigus lahendatakse kaks kontode haldamisega seonduvat probleemi. Esiteks pole personaliosakonnal, kellel on vajadus töötajate andmeid muuta, selleks võimalust, ilma et nende kontodele tuleks süsteemiadministraatori õiguseid anda. Teiseks pole ettevõttel keskkonda, kus oleks näha kõik töötajad koos nende avalike andmetega, list tulevastest sünnipäevadest ning hiljuti liitunud töötajatest.

Antud töö käigus loodi SharePointi *web part*, mille andmevahetuse eest vastutavad kaks Azure *Function*'it ning andmeid hoitakse SharePointi listis. Personalitöötajatele antakse ligipääs vaid SharePointis olevale töötajate listile ja sealt saavad nad hallata kõigi töötajate kontaktandmeid. Rohkem õigusi töötajate kontodele lahenduse kasutamiseks anda ei tule. Kokku moodustavad *web part*, Azure *Function*'id ja SharePointi list tervikliku lahenduse, kus personalitöötajad saavad hallata kõikide töötajate andmeid ning kõik töötajad näevad listi tulevastest sünnipäevadest, uutest töötajatest ning töökaaslaste profiilidest. Iga töötaja saab muuta enda konto andmeid. Nii *web part*'ist kui ka SharePointi listist muudetud andmed sünkroniseeritakse kõikidesse Microsoft Office'i rakendustesse.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 33 leheküljel, 4 peatükki, 24 joonist.

## **Abstract**

### **Creating Azure AD Integrated Employee Contact Management App for SharePoint**

The purpose of this thesis is to create employee contact management app for a company, where everyone has an Azure AD account. There are two problems to be solved during this thesis. Firstly, personnel department, who has the greatest need to manage other workers contact information, does not have a power to do that. It will be only possible, when giving them system administrator rights, but that is not wanted by the company. Secondly, company does not have an application, where are list of all persons who work in that company with everyone's public information. Also there is not a list of upcoming birthdays and recently joined employees.

The outcome of this thesis is SharePoint web part, which data is managed by two Azure Functions and data is stored in a SharePoint list. Personnel employees will have an access to that SharePoint workers list and from there they can manage employees contact information. There is no need to give more rights to employees to use this solution. Web part, Azure Functions and SharePoint list form a complete solution, where personnel employees can manage other employees contact information and all workers can see a list of upcoming birthdays, a list of new workers and profile information of every coworker. Every worker can manage its own profile from web part. Data changed from web part and SharePoint workers list is synchronized to all Microsoft Office applications.

The thesis is in Estonian and contains 33 pages of text, 4 chapters, 24 figures.

## Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> . Rakendusliides, mis võimaldab ühel rakendusprogrammil teise programmi teenuseid kasutada [1]
Azure AD	<i>Azure Active Directory</i> . Microsofti pilvepõhine identiteedi ja ligipääsu haldusteenus [2]
<i>Backend</i>	Tagarakendus, mis tegeleb andmetega ning toetab kasutajaliidest [3]
<i>ClientId</i>	Azure AD rakenduse avalik identifikaator [4]
<i>ClientSecret</i>	Azure AD rakenduse privaatne identifikaator [4]
CORS	<i>Cross-origin resource sharing</i> , allikateülene ressursside jagamine
<i>Deploy</i>	Juurutama, implementeerima
<i>Endpoint</i>	Konkreetne URL, kust API saab vajaliku ressursi
<i>Extension</i>	Tarkvara lisa/laiendus [3]
<i>Frontend</i>	Kasutajaliides. Rakenduse pool, millega suhtleb kasutaja [3]
GET päring	Päring, mille käigus küsitakse andmeid
<i>GraphClient</i>	Tööriist, mis lihtsustab Microsoft Graph'iga päringute tegemist
HTTP	<i>HyperText Transfer Protocol</i> . Andmevahetusprotokoll veebis dokumentide vahetamiseks [3]
JSON	<i>JavaScript Object Notation</i> . Andmevahetusformaad, mis koosneb objektidest, mis omakorda koosnevad nimi-väärtus paaridest [5]
<i>Persona</i>	Office UI Fabric disainielement [6]
POST päring	Päring, mille käigus saadetakse andmeid
<i>Read-only</i>	Kirjutuskaitstud. Andmed, mida saab lugeda, aga ei saa muuta [3]
<i>RedirectUri</i>	URI, kuhu kasutaja suunatakse pärast kasutaja autentimist
<i>TenantId</i>	Azure AD kausta identifikaator
<i>Token</i>	Luba mingit ressursi kasutada [7]
<i>Trigger</i>	Tegevus, mis põhjustab protseduuri automaatse käivitumise [3]

URI	<i>Uniform Resource Identifier</i> , ühtne ressursiidentifikaator
URL	<i>Uniform Resource Locator</i> , ühtne ressursilokaator
<i>Web part</i>	SharePointi lehe komponent [8]
<i>Wireflow</i>	<i>Wireframe</i> ’ide vaheline suhtlus [9]
<i>Wireframe</i>	Rakenduse vaadete struktuur disainita [10]

# Sisukord

1 Sissejuhatus .....	10
1.1 Taust ja probleemistik .....	10
1.2 Töö eesmärgid .....	10
1.3 Metoodika.....	10
1.4 Töö ülesehitus.....	11
2 Analüüs.....	12
2.1 Visioon.....	12
2.2 Nõuded.....	12
2.2.1 Disain.....	15
2.3 Olemasolevad lahendused .....	16
2.3.1 Microsoft Delve.....	16
2.3.2 <i>My account</i> .....	17
2.3.3 Microsoft People .....	19
2.3.4 Azure AD.....	19
2.3.5 SharePointi administraatori keskkond.....	20
2.3.6 Kokkuvõte ja järeldused olemasolevatest lahendustest.....	21
2.4 Arhitektuur ja komponendid.....	21
2.4.1 Lõplik tehniline lahendus .....	21
2.4.2 Olemasolevad <i>web part</i> 'id.....	22
2.4.3 <i>Web part</i> 'i arendus .....	23
2.4.4 Microsoft 365's olevate andmete pärimine .....	23
2.4.5 API valik.....	25
2.4.6 <i>Azure Functions</i> .....	26
3 Lahenduse teostamine.....	27
3.1 SharePointi list.....	27
3.2 <i>Azure Functions</i> .....	27
3.2.1 <i>Azure Function</i> 'i jaoks keskkonna seadistamine .....	28
3.2.2 <i>Function</i> SharePointi listiga suhtlemiseks.....	30
3.2.3 <i>Function</i> andmete sünkroonimiseks .....	32
3.3 <i>Web part</i> 'i loomine.....	34
3.3.1 Struktuur .....	34
3.3.2 Andmed .....	37

4 Tulemused .....	39
4.1 Valminud lahendus .....	39
4.2 Edasiarendus .....	41
Kokkuvõte .....	42
Kasutatud kirjandus .....	43
Lisa 1 - Programmikood .....	47
Lisa 2 – Desktop vaated UI .....	48
Lisa 3 – Graph API'ga kasutaja kohta päritavad andmed .....	50



## Jooniste loetelu

Joonis 1. Kasutusjuhud.....	14
Joonis 2. Desktop vaadete <i>wireflow</i> .....	15
Joonis 3. Delve rakenduses profiili uuendamise vaade [16] .....	16
Joonis 4. My account profiil [16] .....	18
Joonis 5. My account väljade muutmise vaade [16].....	18
Joonis 6. People rakenduse vaade [15].....	19
Joonis 7. Azure AD väljad kasutajaprofiili kohta [18].....	20
Joonis 8. Kasutaja väljade muutmise vaade, osaline [19] .....	20
Joonis 9. Olemasolev sünnipäevade <i>web part</i> .....	22
Joonis 10. <i>People web part</i> .....	23
Joonis 11. <i>GraphClient</i> 'i päringu parameetrid .....	31
Joonis 12. Päring <i>GraphClient</i> 'iga .....	31
Joonis 13. <i>ListItem</i> 'ist andmete lugemine .....	31
Joonis 14. <i>Power Automate</i> 'i töövoogu lisamine listile .....	32
Joonis 15. Microsoft <i>Power Automate</i> 'i loomine .....	33
Joonis 16. <i>Web part</i> 'i struktuur .....	34
Joonis 17. Kasutajate list otsinguga.....	35
Joonis 18. Töötaja enda profiil .....	36
Joonis 19. Enda andmete muutmise vaade .....	36
Joonis 20. Sünnipäevade <i>web part</i> .....	37
Joonis 21. Uute töötajate <i>web part</i> .....	37
Joonis 22. <i>Web part</i> 'i vaade uute töötajate ja sünnipäevadega .....	39
Joonis 23. <i>Web part</i> 'i vaade töökaaslase profiili vaadates.....	39
Joonis 24. Enda profiili muutmise vaade .....	40

# 1 Sissejuhatus

Käesoleva lõputöö teema on „Azure AD'ga integreeritud keskkonna loomine töötajate kontaktide haldamiseks SharePointis“.

## 1.1 Taust ja probleemistik

Kõigil ettevõtte Uptime töötajatel on olemas Azure AD (*Active Directory*) konto, mida kasutatakse erinevatesse keskkondadesse nagu Outlook ja Azure sisselogimiseks. Seoses sellega on ettevõttel kaks lahendust vajavat probleemi:

- 1) Hetkel saab töötajate andmeid muuta vaid süsteemiadministraator ja iga töötaja ise. Personaliosakonnal, kellel on tegelik vajadus töötajate andmeid muuta, selleks võimalust pole ning samuti ei soovita neile anda liiga palju süsteemiadministraatori õiguseid.
- 2) Ettevõttel pole kasutajaliidest, kus oleks list kõigist töötajatest koos nende avalike andmetega, tulevastest sünnipäevadest ning hiljuti liitunud töötajatest.

## 1.2 Töö eesmärgid

Lõputöö eesmärgid on:

- 1) Selgitada välja loodava lahenduse nõuded ja olemasolevate lahenduste sobivus vastavalt nõuetele.
- 2) Valida välja komponendid ja realiseerida lahendus eelpool toodud probleemidele.

## 1.3 Metoodika

Töö metoodika on välja selgitada ettevõtte vajalikud nõuded, võrrelda juba olemasolevaid Microsofti kasutajate haldamise lahendusi ning tuua välja nende sobivus antud probleemidele. Järgneb tehniline analüüs, kus selgitatakse välja lahenduse jaoks kasutatavad komponendid. Töö praktilise osana luuakse SharePointi *web part* (SharePointi lehe komponent), kaks Azure *Function*'it ning SharePointi list.

## 1.4 Töö ülesehitus

Lõputöö koosneb kahest sisupeatükist ning tulemuste peatükist. Esimeses sisupeatükis selgitatakse välja nõuded loodavale lahendusele ning pannakse paika komponendid, mida lahenduse teostamiseks kasutatakse. Teises sisupeatükis kirjeldatakse arendusprotsessi, millest peamiselt keskendutakse Azure *Function*'ite loomisele ning ligipääsuõiguste seadistamisele. Samuti kirjeldatakse *web part*'i struktuuri ja Azure *Function*'iga toimuvat andmevahetust. Tulemuste osas antakse ülevaade loodud lahendusest ning esialgsete eesmärkide täitmisest.

## 2 Analüüs

Analüüsi osa alguses kirjeldatakse antud lõputöö visiooni ja eesmärki. Seejärel selgitatakse välja ettevõtte nõuded loodavale lahendusele ja toetudes olemasolevatele dokumentatsioonidele pannakse paika ka piirangud. Võttes aluseks välja selgitatud nõuded, võrreldakse juba olemasolevaid lahendusi antud valdkonnas ning tuuakse välja nende sobivus ja puudujäägid. Samuti pannakse paika rakenduse lõppvaadete visuaalne disain ning lähtuvalt kõigest eeltoodust valitakse antud probleemile sobiv lahendus.

### 2.1 Visioon

Töö visioon on luua ettevõttele, mis kasutab Azure AD'd, rakendus, kus ettevõtte personaliosakonna töötajad saavad hallata kõigi töötajate kontaktandmeid, mis seejärel salvestuksid Azure AD töötaja konto profiilis. Azure AD's olevat informatsiooni kasutavad ka teised Microsofti rakendused nagu Outlook ja SharePoint. Rakenduse visuaalses osas peab olema nimekiri kõigist ettevõtte töötajatest tööle tuleku järjekorras, list tulevastest sünnipäevadest ja uutest töötajatest. Töötaja profiilile vajutades avaneb töötaja profiil kõigi avalike andmetega. Iga töötaja peab saama muuta enda andmeid ja personaliosakonna töötajad peavad saama muuta kõikide töötajate andmeid.

### 2.2 Nõuded

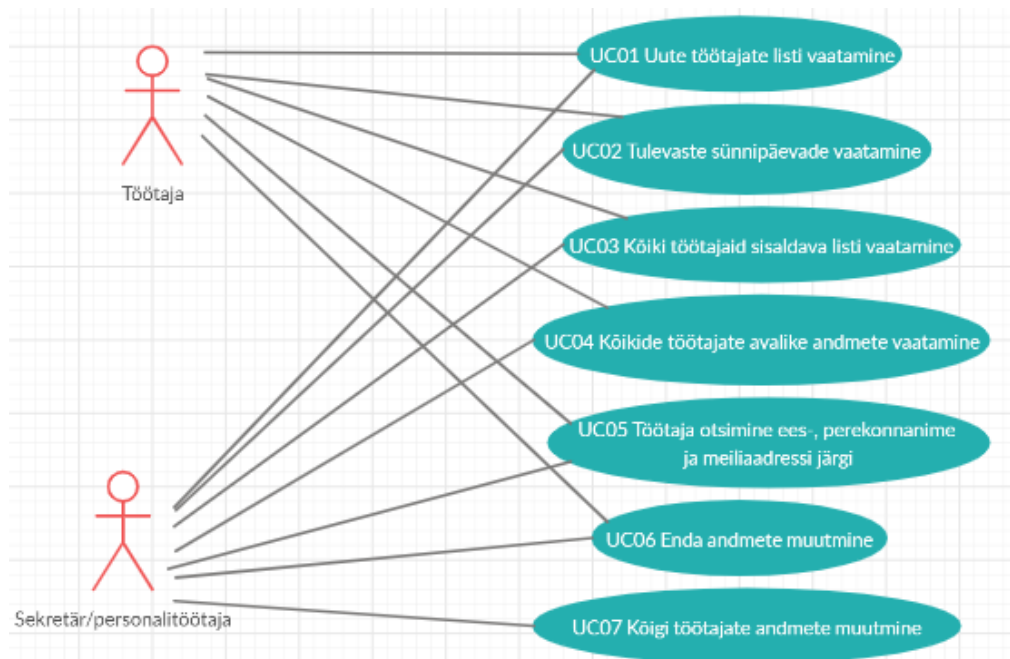
Nõuete kogumiseks suheldi ettevõtte tehnoloogiajuhiga ning sekretäriaga, sest ülesanne pärines esimeselt ja mõlemal on otsene vajadus loodavale rakendusele. Nõudeid täpsustati korduvalt ning täiendati nii prototüübi loomise käigus kui ka erinevate piirangute ilmnemisel.

Mittefunktsionaalsed nõuded prioriteetsuse järjekorras:

1. Informatsiooni presenteerimiseks tuleb luua SharePointi *web part*.
2. *Web part*'is peab olema nimekiri kõigist töötajatest nii, et uued töötajad on listis üleval pool.
3. *Web part*'is on nimekiri uutest töötajatest.

4. *Web part*'is on nimekiri tulevatest sünnipäevadest koos nime ja kuupäevaga.
5. Töötajate nimekirjal peab olema otsinguriba.
6. Personalitöötajad peavad nägema töötajate kohta välju: eesnimi, perekonnanimi, sünnipäev, mobiiltelefon, ametinimetus, meiliaadress, kontori asukoht, foto, kommentaar, LinkedIni konto, Skype'i kasutaja, tööleasumise kuupäev, töölt lahkumise kuupäev, aktiivne (kas kasutaja töötab ettevõttes).
7. Väljad, mida näeb iga töötaja teise töötaja kohta: eesnimi, perekonnanimi, sünnipäev, mobiiltelefon, ametinimetus, meiliaadress, kontori asukoht, foto, kommentaar, LinkedIni konto, Skype'i kasutaja, tööleasumise kuupäev.
8. Lehe disain peab olema sama Office 365 disainiga (kasutatakse Microsoft Fabric UI disainielemente).
9. Leht peab olema inglise keeles.
10. Iga töötaja puhul on nimekirjas kohe välja toodud eesnimi, perekonnanimi ja meiliaadress.

Funktsionaalsetest nõuetest, kus osapooleks on töötaja või personalitöötaja, loodi kasutusjuhtude diagramm. Diagrammi loomiseks kasutati veebirakendust *Creately* [11] (Joonis 1).



Joonis 1. Kasutusjuhud

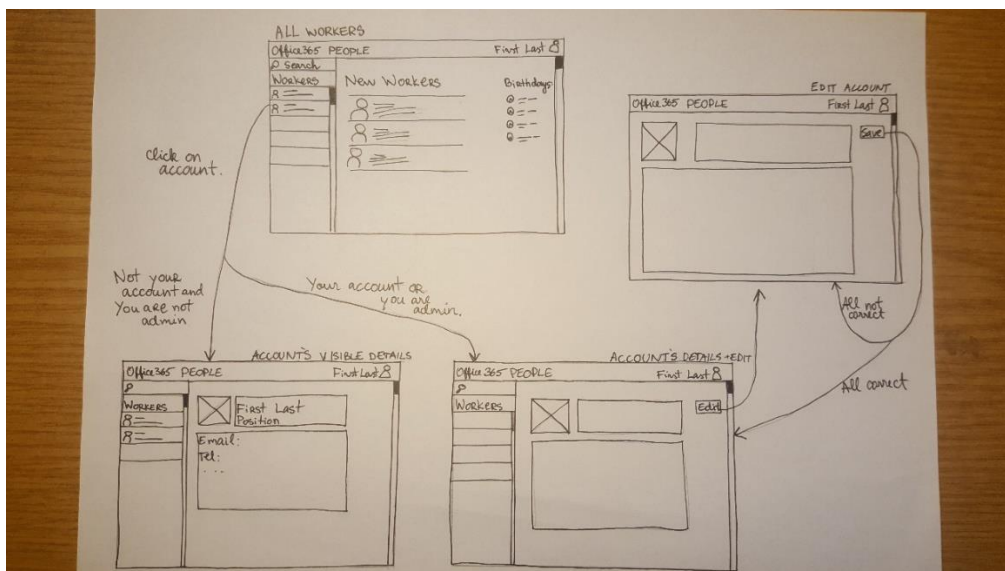
Kõik funktsionaalsed nõuded prioriteetsuse järjekorras:

1. UC07 Ettevõtte personaliosakond saab kõikide töötajate kontaktandmeid muuta.
2. Töötaja nime peale vajutades avanevad desktopil töötajate nimekirja kõrval valitud töötaja kontaktandmed.
3. UC04 Kõik töötajad näevad kõikide töötajate avalikke andmeid.
4. UC06 Töötaja saab enda profiili andmeid muuta.
5. Iga töötaja kontaktandmed on seotud tema Azure AD kontoga.
6. Outlookis töötaja profiili all on näha need samad andmed, mis uues rakenduses on sisestatud.
7. UC03 Kõik töötajad näevad listi ettevõtte töötajatest.
8. UC02 Kõik töötajad näevad tulevasi sünnipäevi.
9. UC01 Kõik töötajad näevad listi hiljuti lisandunud töötajatest.
10. UC05 Otsing otsib töötajaid nende ees- ja perekonnanime ning meiliaadressi järgi.

## 2.2.1 Disain

Antud rakenduse puhul kasutatakse kasutajakogemuse prototüüpimiseks paberit ja pliiaitsit, sest see on kiireim viis panna esialgne visioon paberile ning seda on väikeses tiimis lihtsaim täiendada. Prototüüpimise jaoks otsiti internetist erinevaid SharePointi, People'i ja teiste Office'i rakenduste vaateid ning Office UI Fabric disainielemente [6], sest lõpptulemus peab disainilt olema Office 365'ga võimalikult sarnane.

Esialgu pandi paika detailide paiknemine vaadetes (*wireframe*) [12] ning seejärel liikumine nende vahel. Loodud prototüüpi (Joonis 2) nimetatakse *wireflow*'ks, mis kujutab endast *wireframe* 'ide vahelist suhtlust [9]. Prototüübi põhjal lõi ettevõtte disainer rakendusele lõppvaated (Lisa 2 – Desktop vaated UI).



Joonis 2. Desktop vaadete *wireflow*

Office UI Fabric React disainielementidest üheks olulisemaks, mida kasutatakse, on *Persona*. *Persona* element kuvab konto omaniku pilti ning valitud andmeid. *Persona* elementi läheb loodavas lahenduses vaja mitmes kohas. Esiteks kuvatakse listides igat töötajat *Persona* 'na ning teiseks kasutatakse *Persona* elementi kasutaja profiili juures. Listides soovitatakse kasutada *Persona* 't suuruses 32, 40 või 48, kuid neis suurustes on võimalik kuvada vaid kahte tekstirida, kuid töö nõuetes seisis, et kuvada tuleb töötaja nime, meiliaadressi ja telefoninumbrit. Pärast ettevõtte tehnoloogijuhiga arutamist otsustati, et loodavas lahenduses kuvatakse töötaja nime ja meiliaadressi, telefoninumbrit ei kuvata [13].

## 2.3 Olemasolevad lahendused

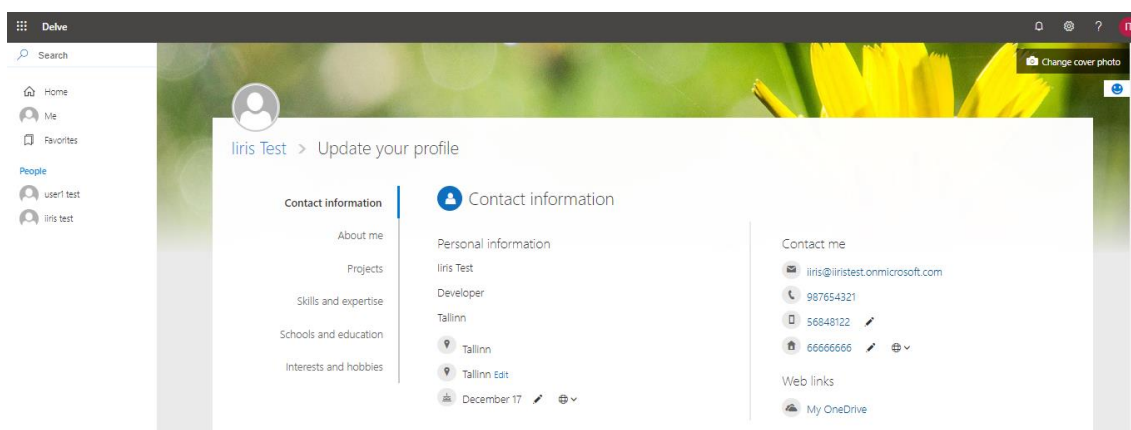
Analüüsi käigus uuriti järgnevaid olemasolevaid lahendusi Microsoft Office 365 keskkonnas, kus saab vaadata või muuta kasutaja andmeid:

- Microsoft Delve,
- *My account*,
- Microsoft People,
- Azure AD,
- SharePointi administraatori keskus.

Selleks, et erinevate Office'i rakenduste administraatori keskkondadesse ligi pääseda, loodi Microsofti arendaja (Microsoft *Developer*) kasutaja, millega saab tasuta ligi kõikidele Microsofti teenustele kolme kuu jooksul [14].

### 2.3.1 Microsoft Delve

Peamisi andmeid nagu ees-, perekonnanimi, meiliaadress ja töö telefoninumber, Delvest muuta ei saa, kuid saab muuta ja lisada täiendavaid kontaktandmeid ning lisada näiteks hobisid ja oskuseid (Joonis 3). Delve keskkond on peamiselt mõeldud organisatsiooni siseselt töötajatele üksteisega failide jagamiseks ja redigeerimiseks [15]. Delves saavad kasutajad enda andmeid muuta samal lehel. Autori katsetuste järel veenduti, et Delves muudetud andmed kajastuvad täielikult SharePointi administraatori keskkuses ning muudetud „minu kohta“ (*About*) informatsioon kajastub ka Office'i People [16] rakenduses.



Joonis 3. Delve rakenduses profiili uuendamise vaade [17]

Delve ei vasta loodava rakenduse nõuetele, sest tavakasutaja ei saa muuta kõiki enda välju ning mis kõige olulisem, pole võimalik anda osadele töötajatele õigusi teiste samas organisatsioonis töötavate inimeste andmeid muuta.



### **2.3.2 *My account***

Microsoft Office'i konto ehk *My account* lehel [18] saab näha enda konto kontaktandmeid (Joonis 4). Muuta (Joonis 5) saab andmeid vaid süsteemiadministraator või kasutaja, kellele on antud spetsiaalsed õigused. Tehes muudatusi *My account* lehel, jõuavad need täielikult Azure AD'sse ja tunni aja jooksul ka SharePointi ning seejärel People ja Delve rakendustesse. Office'i konto lehel ei näe teisi endaga samas organisatsioonis töötavaid inimesi ja seetõttu see keskkond antud töös olevat probleemi ei lahenda.

The screenshot shows a user profile for 'Iiris Test', a Developer in Tallinn. The profile is divided into sections: 'About', 'Address', and 'Contact details'. Each section has an 'Edit' link. The 'About' section lists first name 'Iiris', last name 'Test', occupation 'Developer', and department 'Tallinn'. The 'Address' section lists 'Tänav 5', postal code '12345', city 'tallinn', state 'Harju', and country 'Estonia'. The 'Contact details' section lists email 'iiris@iiristest.onmicrosoft.com', alias 'iiris', mobile '12345678', phone '987654321', and alternate email 'iiristest@outlook.com'.

Joonis 4. My account profiil [18]

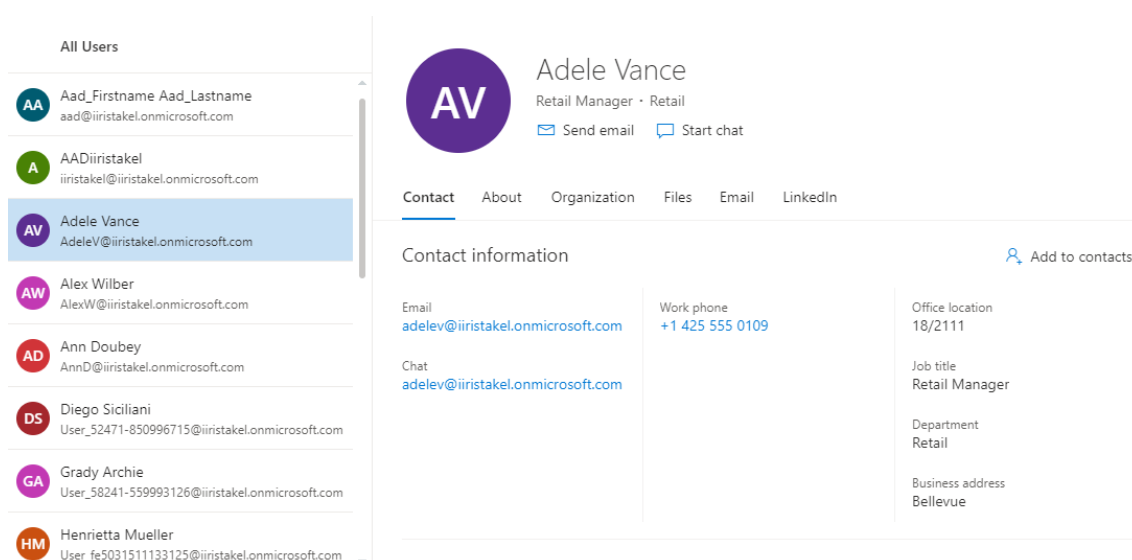
The screenshot shows the edit form for the user profile. It contains input fields for all the information shown in the previous screenshot. The 'About' section has fields for first name, last name, occupation, and department, with 'Save' and 'Cancel' buttons below. The 'Address' section has fields for address, postal code, city, state/province, and country/region, with 'Save' and 'Cancel' buttons below. The 'Contact details' section has fields for email, alias, mobile, phone, and alternate email, with 'Save' and 'Cancel' buttons below. A warning message is present: 'Don't use your Office 365 email address here. Why not?'

Joonis 5. My account väljade muutmise vaade [18]

### 2.3.3 Microsoft People

Microsoft People rakenduses [16] on võimalik avada menüüst kõikide organisatsiooni töötajate vaade (*All Users*), kust näeb kõiki ettevõtte töötajaid (Joonis 6). Vaadates kellegi teise profiili näeb andmeid, mis on sisestatud kas Delvest, SharePointist või Azure AD'st, sest People rakendusest otse andmeid muuta ei saa. Enda profiili kohta näeb sama palju andmeid kui teiste profiilide kohta, kuid on olemas ka nupp „värskenda profiili“, mis viib Delvesse.

Kuigi rakendus on osaliselt üsna sarnane sellele, mida tahetakse luua (list kõigist töötajatest ja valitud töötaja profiili kuvamine), siis sellegipoolest ei sobi see lahendama käesolevas töös esitletud probleemi, sest pole võimalik anda õigusi, et osad töötajad saaksid muuta kõigi töötajate andmeid. Samuti pole võimalik kuvada listi tulevastest sünnipäevadest ja lisandunud töötajatest.



Joonis 6. People rakenduse vaade [16]

### 2.3.4 Azure AD

Azure AD's [19] tehtud muudatused jõudsid SharePointi peale tunnist ootamist. Kontoga seotud „kuvatav nimi“ jõudis People rakendusse, kõik andmed jõudsid aga *My account* lehele ning ligi tunni pärast ka Delve rakendusse.

Azure AD's näeb kõiki ettevõtte töötajaid ja saab vaadata nende kontaktandmeid (Joonis 7). Andmeid saab muuta süsteemiadministraator, aga neid õigusi nii personalitöötajatele kui ka teistele töötajatele ei anta.

## Identity [edit](#)

Name	First name	Last name
Iiris Test	Iiris	Test
User name	User type	
iiris@iiristest.onmicrosoft.com	Member	
Object ID	Source	
626141e9-d8c9-48f6-8357-cc9d332fcab6	Azure Active Directory	

## Job info [edit](#)

Job title	Department	Manager
Developer	Tallinn	<a href="#">user2 test</a>

## Settings [edit](#)

Block sign in	Usage location
No	Estonia

## Contact info [edit](#)

Street address	State or province	Country or region	Office
Tänav 5	Harju	EE	Tallinn
City	ZIP or postal code	Office phone	Mobile phone
tallinn	12345	987654321	12345678

Joonis 7. Azure AD väljad kasutajaprofili kohta [19]

## 2.3.5 SharePointi administraatori keskkond

SharePointis on kõige rohkem erinevaid andmevälju (Joonis 8), mida kasutaja kohta täita saab, kuid neid välju saab muuta vaid süsteemiadministraator [20]. Tavalistel kasutajatel ülevaadet kõigist töötajatest pole, kuna nad ei pääse sellele keskkonnale ligi.

SharePointis muudetud andmed muutuvad ka Delves, kuid People rakenduses muutub vaid „minu kohta“ (*About*) informatsioon ning Azure AD's ja *My Account* lehel ei muutu midagi.

## User Profiles

Use this page to edit this user profile by changing values for the following properties. Properties that are mapped to the external data source will be overwritten the next time user profiles are imported.

Account name:	i:0#.f membership iiris@iiristest.onmicrosoft.com	Show To	Everyone
First name:	<input type="text" value="Iiris"/>	Show To	Everyone
Last name:	<input type="text" value="Test"/>	Show To	Everyone
Name: *	<input type="text" value="Iiris Test"/>	Show To	Everyone
Work phone: *	<input type="text" value="9999999"/>	Show To	Everyone
Department: *	<input type="text" value="Tallinn"/>	Show To	Everyone
Title: *	<input type="text" value="Developer"/>	Show To	Everyone
Department:	<input type="text" value="Tallinn"/>	Show To	Everyone ▾
Manager:	<input type="text" value="user1.test"/>	Show To	Everyone
About me:	<input type="text" value="About me text."/>	Show To	Everyone
Personal site:	<input type="text" value="/personal/iiris_iiristest_onmicrosoft_com/"/>	Show To	Everyone

Provide a personal description expressing what you would like others to know about you.

Joonis 8. Kasutaja väljade muutmise vaade, osaline [20]

### **2.3.6 Kokkuvõte ja järeldused olemasolevatest lahendustest**

Pärast mitmeid katsetusi ning andmete muutmisi olid tulemused järgmised. Administraatori õigustes kasutajaga sai enda andmeid muuta neljas erinevas keskkonnas: Delves, Azure AD's, *My Account* lehel ja SharePointi administraatori keskuses. Kõikide kasutajate andmeid sai muuta kahes keskkonnas – Azure AD's ja SharePointi administraatori keskuses. Tavakasutajana sai enda andmeid muuta vaid Delves.

Andmeid kontode kohta hoitakse peamiselt kahes kohas – Azure AD's ja SharePointis. Delves on olemas kõik profiili väljad, mis SharePointis, kuid välju on SharePointis kokku rohkem. People rakenduse andmed tulid aga mitmest kohast – profiili nimi muutus Azure AD või *My Account* lehe põhjal ning ülejäänud väljad nagu „minu kohta“, koolid, oskused (*about, schools, skills*) muutusid SharePointi või Delve andmete põhjal.

Kasutajate andmed ja nende halduskohad on mitmes erinevas keskkonnas laiali. Üheski eeltoodud kohas ei saa personaliosakonnale ja sekretärile anda õiguseid kõigi töötajate andmete haldamiseks. Samuti on puudu mitmed väljad, mis on loodava lahenduse jaoks vajalikud. Kuna olemasolevad lahendused ei sobi, tuleb luua uus.

## **2.4 Arhitektuur ja komponendid**

Esialgu planeeriti lahendada probleem nii, et töötajate kuvamiseks ja profiilidel andmete muutmiseks luuakse SharePointi *web part*, mis suhtleb Microsoft Graph API'ga (*Application Programming Interface*). Andmeid töötajate kohta päritakse Graph API *User* (kasutaja) objekti käest. Personaliosakond ja sekretär näevad järgmisi välju: eesnimi, perekonnanimi, sünnipäev, mobiiltelefon, ametinimetus, meiliaadress, kontori asukoht, foto, kommentaar, Linkedini konto, Skype'i kasutaja, tööleasumise kuupäev, töölt lahkumise kuupäev, aktiivne (kas kasutaja töötab ettevõttes). Ülejäänud töötajad näevad kõiki välju, välja arvatud töölt lahkumise kuupäev ja aktiivne. Andmetele piiratud ligipääsu andmiseks kasutatakse Azure *Function*'it, mis küsib Graph API *User* objektide käest andmeid ning edastab lõppkasutajale andmed, mida tal on õigus näha.

### **2.4.1 Lõplik tehniline lahendus**

Lahendust realiseerides selgus, et Microsoft Graph API kaudu pole võimalik küsida kõiki töötajaid tööle asumise järgi sorteerituna ning ei saa korraga pärida kõigi kasutajate

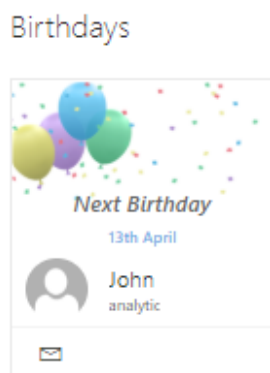
sünnipäevi. Samuti selgus, et *User* objekti küljes pole algupärselt kõiki vajalikke välju, mida antud lahenduse jaoks on.

Seetõttu otsustati muuta esialgset lahendust ja luua SharePointi list ettevõtte töötajatest koos kõikide vajalike väljadega ning kaks Azure *Function*'it. Üks *Function* sünkroonib listis muudetud töötajate andmed Azure AD'sse, et töötajate andmed jõuaks läbi Azure AD kõikidesse Microsoft Office'i rakendustesse. Teine Azure *Function* loodakse selleks, et töötajad saaksid ligi vaid piiratud hulgale SharePointi listi andmetele. Õigused listist kõigi andmete lugemiseks ja sinna kirjutamiseks antakse *Function*'ile ning *Function* edastab *web part*'i vaid need andmed, mida kõik ettevõtte töötajad näha võivad. Iga töötaja saab enda profiilil andmeid muuta loodavas SharePointi *web part*'is.

#### 2.4.2 Olemasolevad *web part*'id

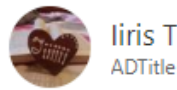
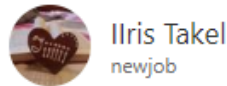
Hetkel on SharePointis olemas mitmeid *web part*'e, mida saab kasutada listi kuvamiseks. Samuti on olemas *web part*'e otsingu ja otsingutulemuste jaoks [21]. Neid saab sobivuse korral loodavas töös kasutada ning vajadusel arendada.

Hetkel on olemas *web part* tulevaste sünnipäevade kuvamiseks (Joonis 9), mille kasutamiseks tuleb luua kas käsitsi SharePointi list sünnipäevadest või kasutada samas projektis olevat Azure *Function*'it, mis loob ise sünnipäevade listi SharePointi [22].



Joonis 9. Olemasolev sünnipäevade *web part*

Uute töötajate kuvamiseks oleks võimalik kasutada *People web part*'i (Joonis 10), kus tuleb käsitsi lisada töötaja. Seejärel saab valida, kas kuvatakse vaid töötaja nimi ja amet või lisatakse suurem kaart, kuhu lehe haldaja saab lisada ka kirjelduse töötaja kohta ning lingi [23].



Joonis 10. *People web part*

Kuigi loodeti osaliselt kasutada olemasolevaid *web part*'e, siis erinevaid *web part*'e uurides selgus, et need siiski ei lahenda probleemi, sest pole *web part*'i, kus saaks töötajaid soovitud järjekorras kuvada. Ka ei leitud *web part*'i, kus saaks töötajate andmeid muuta ning muutmise õiguseid näiteks personaliosakonnale anda. Ka eelnevalt välja toodud sünnipäevade ja *People web part* ei sobi, sest sünnipäevade *web part* presenteerib informatsiooni ebasobival viisil ning see pole muudetav ja *People web part*'e saab vaid käsitsi ära võtta ja uusi panna.

### 2.4.3 *Web part*'i arendus

SharePointi *web part*'i loomiseks on võimalik kasutada erinevaid JavaScript raamistikke, millest enim kasutuses on Angular, jQuery, Knockout, React, Vue ja Handlebars, samuti saab *web part*'i luua ilma raamistikuta [24]. Töö autor oli eelnevalt JavaScriptiga vaid põgusalt kokku puutunud ning vähesel määral oli varem kasutatud Angulari, Reacti ja Vue'd, seetõttu ülejäänud valikud eemaldati nimekirjast, et ei tuleks täiesti uut raamistikku endale selgeks tegema hakata.

Valik tehti selle järgi, et vaid Reactiga *web part*'i arendades tulevad automaatselt kaasa Office UI Fabric React disainielemendid [25], mis oli loodava lahenduse üks nõuetest. AngularJS'il on küll olemas sarnane disainielementide komplekt, kuid seda enam edasi ei arendata, sest enamuse arendusi on AngularJS'i pealt liikunud uuemate Angulari versioonide peale [26]. Saab kasutada ka Office UI Fabric JS'i, mis on mõeldud kõikidele JavaScript raamistikele, kuid seda samuti enam ei toetata ja edasi ei arendata ning soovitatakse kasutada just Office UI Fabric Reacti [27].

### 2.4.4 Microsoft 365's olevate andmete pärimine

Rakenduses andmete pärimiseks ja muutmiseks tuleb kasutada API'sid. API'de kaudu saab küsida näiteks listi ettevõtte töötajatest, andmeid ühe töötaja kohta või hoopis uuendada andmeid konkreetse kasutaja puhul. Azure AD kontode kohta andmete

saamiseks on olemas mitmeid API'sid. Esialgu uuritakse Microsoft Graph API'sid, sest lahendus luuakse Azure AD kontode jaoks ning just Graph API'dest saab andmeid kontode kohta [28]. Samuti uuritakse SharePointi API'sid, kuna lahendus luuakse SharePointis kasutamiseks [29].

Kasutades API't „<https://graph.microsoft.com/v1.0/users>“ tagastatakse JSON (*JavaScript Object Notation*) kujul list ettevõtte töötajatest järgnevate andmetega: töö telefonid, kuvatav nimi, eesnimi, id, ametinimetus, meiliaadress, mobiilnumber, kontori asukoht, eelistatud keel, perekonnanimi ja kasutajanimi (*businessPhones, displayName, givenName, id, jobTitle, mail, mobilePhone, officeLocation, preferredLanguage, surname and userPrincipalName*). Kui kasutada lisaparaameetrit „\$select“, saab määrata, milliseid andmeid päritavate objektide kohta tagastatakse. Pääringu „[https://graph.microsoft.com/v1.0/users?\\$select=displayName,surname,jobTitle](https://graph.microsoft.com/v1.0/users?$select=displayName,surname,jobTitle)“ puhul tagastatakse list kasutajatest, kus iga kasutaja juures on ära toodud tema kuvatav nimi, perekonnanimi ja ametinimetus [30]. Kõik erinevad paraameetrite võimalused on välja toodud Microsoft Graph'i dokumentatsioonis [31]. Graph API tagastab tulemusi nii Azure AD'st kui ka SharePointist. Kui väljad on samad, näiteks eesnimi, perekonnanimi ja ametinimetus, siis tulevad väärtused Azure AD'st. Lisas 3 on JSON formaadis välja toodud kõik paraameetrid, mida Graph API't kasutades kasutaja kohta on võimalik saada [32]. Tumedamaks on tehtud need väljad, mida on igal tavakasutajal võimalik ise uuendada [33]. Graph API'ga muudetud kasutaja andmed kajastuvad nii Azure AD's, *My Account* lehel kui ka SharePointis ja Delves ning sealt edasi sünkroonitakse andmed ka People rakendusse.

SharePointi API'de kaudu saab teha päringuid andmete kohta, mis asuvad SharePointi saidil. Seega saab SharePointi API kaudu pärida kasutajate kohta andmeid vaid siis, kui on tehtud kasutajate andmetest list SharePointi lehele. Listist andmete pärimiseks saab kasutada järgmist API't: „[,<siteUrl>/\\_api/Lists/getbytitle\('listname'\)/items](https://graph.microsoft.com/v1.0/sites/<siteUrl>/_api/Lists/getbytitle('listname')/items)“, kus *listname* on listi nimetus ja *<siteUrl>* on SharePointi sait. Samuti saab pärida SharePointi API kaudu andmeid sisseloginud kasutaja ehk enda profiili kohta, kasutades API't „[,<siteUrl>/\\_api/SP.UserProfiles.PeopleManager/getmyproperties](https://graph.microsoft.com/v1.0/sites/<siteUrl>/_api/SP.UserProfiles.PeopleManager/getmyproperties)“. SharePointil on olemas ka otsingu API „[,<siteUrl>/\\_api/search/query](https://graph.microsoft.com/v1.0/sites/<siteUrl>/_api/search/query)“, mis otsib üle kasutajaga seotud SharePointi lehekülgede ja neis olevate dokumentide [34].



## 2.4.5 API valik

Kui esialgu taheti kasutada Microsoft Graph API't, et küsida kasutaja kohta andmeid, tehes päringuid kasutaja (*User*) objekti kohta, siis pärast Graph API'ga lähemat tutvumist selgus, et Graph API'ga ei saa tööleasumise kuupäeva (*hireDate*) järgi kasutajaid sorteerida ning küsida korraga kõigi töötajate sünnipäevi. Samuti pole Graph API *User* objekti küljes algupäraselt kõiki nõuetes välja toodud välju – sellisteks väljadeks on Linkedini konto, Skype'i kasutaja, kommentaar, töölt lahkumise kuupäev ja aktiivne (kas kasutaja on ettevõtte töötaja). Samuti pole kõiki vajalikke välju võimalik saada SharePointi API'ga, sest ka sellega pole antud välju võimalik ilma lisa tabelleid loomata pärida. Seega tuleb otsustada, kuidas rakenduse arhitektuuri muuta, et saada lahenduse jaoks kõik vajalikud listid ning andmeväljad.

Üheks võimaluseks oleks lisaväljad eraldi SharePointi listi panna, kuid siis tuleks luua personaliosakonnale eraldi kasutajaliides, kus Microsoft Graph API *User* objektiga seotud välju muuta ning siis peaksid ettevõtte personalitöötajad käima andmeid kahes erinevas kohas muutmas. Teiseks võimaluseks oleks luua SharePointi list töötajatest koos kõigi vajalike väljadega, et oleks üks konkreetne koht, kust andmeid pärida ja kuhu salvestada. Et SharePointi listis muudetud andmed ka Graph API'sse jõuaksid, saab luua Azure *Function*'i. SharePointi listi loomine lahendaks ka töö peamise probleemi, et personaliosakonnal ei ole võimalik kõigi kasutajate andmeid muuta, ilma et nende Azure AD kontodele lisaõiguseid anda. SharePointi listis töötajate andmete muutmiseks tuleb personalitöötajatele anda vaid ligipääs loodud listile [35]. Õigused töötajate kontodel andmeid muuta antakse hoopis Azure *Function*'ile, kus on ka ära määratud, milliseid andmeid töötajate kontodel muuta saab.

Eelnevat arvesse võttes luuakse SharePointi list töötajatest koos kõigi vajalike väljadega. Samuti lahendab SharePointi listist andmete pärimine probleemi, kuidas saada list ettevõtte töötajatest tööle asumise järjekorras, kuna SharePointi listi veergudele on võimalik lisada indekseid ja nende järgi andmeid sorteerida [36].

SharePointi listist saab andmeid pärida kasutades nii eelnevalt välja toodud SharePointi API't kui ka Microsoft Graph API't. Loodavas lahenduses kasutatakse Graph API't. Graph API on uuem ning seda arendatakse pidevalt [28]. Samuti on Graph API'ga ligipääs suuremale hulgale ressurssidele [37].

#### 2.4.6 Azure *Functions*

Azure *Functions* töötab loodava lahenduse *backend*'ina (tagarakendus, mis tegeleb andmetega ning toetab kasutajaliidest). Kõik andmed, mida rakendus kasutab, käivad sealt läbi. Seda seetõttu, kuna SharePointi listile pole veel võimalik lisada veerupõhist ligipääsu - kirjeldatakse vaid ligipääsu tervele listile, osadele ridadele või mitte millelegi [35]. Isegi kui loodava rakenduse kaudu peaks iga töötaja nägema vaid osaliselt seda informatsiooni, mis listis on, siis tegelikult peab talle andma õiguse näha kogu informatsiooni ja seda ettevõtte ei soovi. Lahenduseks ongi õiguste andmine Azure *Function*'ile [38], millel on õigus küsida kõiki andmeid listist, kuid edasi *frontend*'ile (rakenduse pool, millega suhtleb kasutaja) annab ta API'na vaid selle osa, mida rakendusel on tarvis kuvada ning nii ei tule töötajate Azure AD kontodele üldse lisaõigusi anda. Ka andmete muutmine toimub läbi *Function*'i. Juba päritud väljade muutmisele paneb piirangu *frontend*. Nagu listist andmete lugemise puhul, ei anta listi kirjutamiseks õiguseid töötajatele, vaid õigused antakse *Function*'ile.

Peale listist andmete küsimise ja sinna salvestamise kasutatakse ka teist Azure *Function*'it, mida läheb tarvis, et SharePointi listis muutunud andmed uueneksid ka teistes Office'i rakendustes. Selleks luuakse *Function*, mis saab sisendiks ainult need kasutaja andmed, mis on Microsoft Graph API *User* objekti küljes olemas. Kasutajakontoga seotud andmeid on võimalik igal pool korraga muuta kasutades Microsoft Graph API't.

## 3 Lahenduse teostamine

Arenduse tulemusena loodakse SharePointi saidi tarvis *web part*, mis kuvab listi töötajatest, valitud töötaja profiili, uusi töötajaid ning tulevasi sünnipäevi. *Web part* luuakse kasutades Reacti. Kuna oli tarvis teha andmemudelid muudatusi võrreldes Microsoft Graph API'ist saadavate andmetega, luuakse SharePointi list töötajatest, mida saavad muuta vaid ettevõtte personaliosakonna töötajad. Andmetele ligipääsu piiramiseks luuakse Azure *Function*. Teine *Function* luuakse selleks, et nii *web part*'is kui ka otse SharePointi listis muudetud andmed muutuksid ka teistes Office'i rakendustes. Päringute tegemiseks kasutatakse Graph API'sid. Programmikood on kättesaadav aadressil, mis on välja toodud Lisas 1.

### 3.1 SharePointi list

Kõigepealt loodi SharePointi saidile list ettevõtte töötajatest. Listi nimeks pandi *Workers* ehk töötajad. Listi lisati kõik veerud, mis analüüsi käigus välja selgitati. Kõik tabelile lisatud veerud on järgmised: *Firstname* (eesnimi), *Lastname* (perekonnanimi), *MobilePhone* (telefoninumber), *HireDate* (tööle asumise kuupäev), *Birthday* (sünnipäev), *JobTitle* (ametinimetus), *Photo* (pilt), *Comment* (kommentaar), *Linkedin* (LinkedIni konto), *LeaveDate* (töölt lahkumise kuupäev), *Active* (aktiivne, kas kasutaja on ettevõtte töötaja), *Skype* (Skype'i kasutaja), *OfficeLocation* (kontori asukoht), *Mail* (meiliaadress), *AADGuid* (Azure AD unikaalne identifikaator).

Lisaks on listil automaatselt olemas veerg *Title* (pealkiri) ning *Id* (identifikaator), mis on igal loodud real unikaalne ja luuakse automaatselt. Samuti on listil alati olemas veerud *Modified* (muudetud), *Created* (loodud), *Created By* (kelle poolt loodud) ja *Modified By* (kelle poolt muudetud), mis annavad teavet selle kohta, kes antud listi rea lõi, muutis ja millal need tegevused toimusid – need veerud täituvad automaatselt ja neid käsitsi muuta ei saa.

### 3.2 Azure *Functions*

Loodava lahenduse jaoks tuleb luua kaks Azure *Function*'it. Üht *Function*'it läheb tarvis andmete pärimiseks SharePointi listist ja sinna salvestamiseks. Teist *Function*'it läheb tarvis, et SharePointi listis muutunud andmed Azure AD'sse ja teistesse Office'i

rakendustesse sünkroonida. Azure *Function*'ite arendamiseks kasutatakse C# programmeerimiskeelt ja andmeid küsitakse ning salvestatakse kasutades Microsoft Graph API't.

### 3.2.1 Azure *Function*'i jaoks keskkonna seadistamine

Azure *Function*'i loomiseks kasutati õpetust Microsofti dokumentatsioonist. Azure *Function* loodi Visual Studio Code'iga, samas projektis, kus loodi *web part*. Visual Studio Code'i saab lisada Azure'i laienduse (*extensions*'i), mille abil on *Function*'e lihtne luua ja kasutusele võtta (*deploy*'da). Seejärel loodi Azure'i laiendusega uus Azure *Functions*'i projekt ning sinna esimene *Function*, mille loomiseks tehti järgmised valikud: programmeerimiskeeleks valiti C#, malliks HttpTrigger, sest loodav *Function* peab käivituma HTTP (*HyperText Transfer Protocol*) päringute peale, ning autoriseerimise leveliks *Function* [39], [40].

Selleks, et hakata loodud *Function*'iga Microsoft Graph API kaudu SharePointi listist andmeid küsima ja salvestama, läheb tarvis *GraphClient*'it (tööriist, mis lihtsustab Microsoft Graph'iga päringute tegemist), mille kaudu päringuid teha. *GraphClient*'i jaoks läheb aga tarvis *token*'it (luba mingit ressursi kasutada), mis päringu vastuses kaasa antakse, kui päringu tegijal on õigus antud päringut teha. *Token*'i saamiseks kasutati koodi kasutaja BrianTJackett'i *BTJ.CSAdvent.AZFunc*-nimelisest githubi repositooriumist [41]. *Token*'i saamiseks läheb tarvis rakenduse *tenantId*'d (Azure AD kausta identifikaator), *clientId*'d (Azure AD rakenduse avalik identifikaator), *clientSecret*'it (Azure AD rakenduse privaatne identifikaator) ja *redirectUri* (URI, kuhu kasutaja suunatakse pärast kasutaja autentimist). Need kõik väärtused saab Azure AD'st, kui sinna uus funktsiooni rakendus (*Function App*) luua. Seega tuleb kõigepealt vastloodud *Function* Azure'i *deploy*'da. Selle jaoks saab taas kasutada Azure'i laiendust Visual Studio Code'is. Selleks, et *Function*'it *deploy*'da, läheb vaja Azure'i tellimust (*Subscription*'it), sest muidu ei saa Azure'i teenuseid kasutada. *Function*'i esmakordsel Azure'i avaldamisel tulebki luua Azure'i uus funktsiooni rakendus. See aitab hiljem erinevaid *Function*'e kergemini hallata, kui need on jaotatud erinevate rakenduste vahel. Edasi tuleb teha veel mitmeid valikuid - valida asukoht enda loodavatele ressurssidele ja anda funktsiooni rakendusele unikaalne nimi [39], [42].

Kui funktsiooni rakendus on loodud ja *Function* on Azure'i üles laetud, tuleb minna Azure'i portaali [19]. Selleks, et loodud funktsiooni rakendust ei saaks igaüks kasutada,

muudetakse rakenduse autentimise ja autoriseerimise seadeid. Selleks tuleb Azure'i portaalist valida *Function App* ning sealt enda loodud rakendus. Seejärel tuleb avada *Platform features* (platvormi omadused) ning sealt *Authentication/Authorization* (autentimine/autoriseerimine). Autentimisel tuleb teha valik „On“ ja seejärel seadistada, mis peab juhtuma, kui kasutaja püüab rakendusele ligi saada, aga pole autenditud. Kuna loodud lahenduse käigus kasutatakse Azure AD kontosid, suunatakse autentimata kasutaja sisse logima Azure AD kaudu. Tehakse valik *Log in with Azure Active Directory* (logi sisse Azure AD'ga). Azure AD autentimise pakkujana (*authentication provider*) seadistamiseks tuleb valida *Azure Active Directory*, seejärel valida *Express, Create new AD App* (loo uus AD rakendus) ja valida nimi uuele rakendusele. Selle tegevuse käigus luuakse Azure AD'sse uus *App Registration* (registreeritud rakendus) ja varem loodud funktsiooni rakendus (*Function App*) saab seda kasutada kui autentimise pakkujat [43].

Nüüd saab kõik vajalikud andmed rakenduse jaoks minnes Azure AD'sse. Menüüst tuleb valida *App registrations*, ning seejärel valida loodud rakendus. Rakenduse alt saab kätte *tenantId*, *clientId* ja *redirectUri*. *ClientSecret* tuleb esmalt luua - selleks tuleb menüüst valida *Certificates & secrets* (Sertifikaadid ja salasõnad) ja sealt luua uus *clientSecret*. See tuleb kuhugi üles kirjutada, kuna Azure'st selle väärtust hiljem enam näha ei saa. Sellega on kätte saadud kogu vajalik informatsioon *token*'i küsimiseks [38].

Kuna loodavatel *Function*'itel peab olema õigus teha eelnevalt väljatoodud päringuid, tuleb järgmiseks lisada *Function*'e sisaldavale rakendusele õigused. Selleks tuleb *App registrations* lehel enda loodud rakenduse alt valida *API permissions* ehk API õigused. Uute õiguste lisamiseks tuleb valida *Add a permission* ehk lisa õigus. Kuna loodavas lahenduses kasutatakse Microsoft Graph API't, tuleb õigusi lisada Microsoft Graph'ile. Seejärel tuleb valida, mis tüüpi õiguseid on rakendusel vaja, kas õigused lisatakse kasutajatele ja päringuid tehakse kasutajate nimel või õigused lisatakse rakendusele ja päringuid tehakse rakenduse nimel. Kuna antud lahenduse jaoks on vaja kasutajate õigusi nii laiendada kui ka samal ajal piirata, tuleb õigused lisada rakendusele ja teha päringud rakenduse õigustes. Õigused, mida loodavate *Function*'ite jaoks läheb tarvis: *Sites.ReadWrite.All*, mis lubab SharePointi saitidelt (sealhulgas listidest) andmeid lugeda ja sinna kirjutada, *User.ReadWrite.All*, mis lubab Azure AD kasutajate andmeid lugeda ja muuta [33], [44].

Veel tuleb Azure's funktsiooni rakenduse alt seadistada CORS (*Cross-Origin Resource Sharing*). CORS tuleb seadistada seetõttu, kuna *web part* asub SharePointis ning Azure *Function* asub Azure's. CORS võimaldab erinevate hostide vahelist suhtlust. Et SharePointis *frontend*'ina tegutsev *web part* pääseks ligi *backend*'ina tegutsevale Azure *Function*'ile, lisatakse Azure's CORS'i seadetes lubatud hostina töö käigus kasutatav SharePointi leht [45].

### 3.2.2 *Function* SharePointi listiga suhtlemiseks

Andmeid, mida rakendus kasutab, hoitakse SharePointi listis. Kuid listis on ka välju, mida kõigile töötajatele välja näidata ei taheta. Hetkel on sellisteks väljadeks töölt lahkumise kuupäev ja aktiivne (kas kasutaja töötab ettevõttes), kuid neid välju võib tulevikus lisanduda veelgi. Kuna SharePointi listis pole hetkel võimalik anda veerupõhist ligipääsu, saavad kasutajad ligi kas kõigile veergudele või mitte ühelegi. Antud lahenduses otsustati, et töötajad, kes ei kuulu personaliosakonda, ei saa listile üldse ligi ja õigused listist andmete küsimiseks ja salvestamiseks antakse hoopis Azure *Function*'ile, mis annab lõppkasutajale API'na edasi vaid talle näha olevad väljad.

*Function*'il *GetAndUpdateUsersFunc* (Lisa 1 - Programmikood) on sisendiks *HttpRequest*, mis sisaldab tehtud HTTP päringut. Selleks, et saada andmeid kõikide töötajate kohta, tuleb teha HTTP GET päring, parameetrite ja kehatekstiga. Seejärel kasutatakse loodud *GraphClient*'it, et Microsoft Graph API kaudu SharePointi listist kõik aktiivsete kasutajate read kätte saada (Joonis 12). Kõigi vajaminevate veergude lugemiseks tuleb päringule lisada parameetrid *expand* (laienda) ja *select* (vali). Nende parameetriteta tagastataks vaid listi metaandmed. Lisades *expand* parameetri koos valikuga *fields* (väljad), tagastatakse ka kõik listi kirjed koos kõigi veergudega. *Fields* parameetrit aga täiendatakse *select* valikuga, sest küsitakse vaid konkreetset hulka andmeid (Joonis 11).

```
List<QueryOption> options = new List<QueryOption>
{
    new QueryOption("expand", "fields(select=Id,AADGUID,Firstname,Lastname,Mail,JobTitle,MobilePhone," +
        "OfficeLocation,HiringDate,Birthday,Skype,Linkedin,Comment)")
};
```

Joonis 11. *GraphClient*'i päringu parameetrid

```
var usersList = await graphClient.Sites["{spSiteId}"].Lists["{spListName}"].Items
    .Request(options)
    .Filter("fields/Active eq 1")
    .OrderBy("fields/HireDate desc")
    .GetAsync();
```

Joonis 12. Päring *GraphClient*'iga

Kuna Graph API'ga SharePointi listist andmeid lugedes tagastatakse kollektsoon *ListItem* itest (tabeli ridadest), siis tekkis esialgu probleem, kuidas igast kirjest saada kätte kõik vajalikud väljad kasutaja kohta. Peale mitmeid katsetamisi leiti olukorra lahenduseks järgnev lahendus (Joonis 13), kus *user* on eelmainitud *ListItem*:

```
birthday = user.Fields.AdditionalData["Birthday"]?.ToString(),
comment = user.Fields.AdditionalData["Comment"]?.ToString(),
```

Joonis 13. *ListItem*'ist andmete lugemine

Selleks, et *Function*'i poolt tagastatavad andmed sisaldaksid vaid neid väljasid, mida *web part* il vaja läheb, loodi uus mudel *IUser* ning pärast SharePointi listist andmete lugemist tagastabki *Function* API'na listi *IUser* objektidest. *Function* tagastab SharePointi listist read, kus kasutajale on märgitud „aktiivne“ ning sorteerib need tööle asumise järjekorras nii, et uued töötajad oleksid näha esimestena.

Kui tahetakse loodud Azure *Function*'ilt saada vaid ühe kasutaja andmeid, tuleb päringusse kaasa anda parameeter *Mail* ehk meiliaadress ning seejärel *GraphClient*'iga tehtava päringu käigus filtreeritakse välja esimene kasutaja, kus veerg *Mail* on võrdne päringusse lisatud meiliaadressiga. Selleks aga, et SharePointi listis konkreetse veeru järgi andmeid filtreerida, tuleb sellele veerule lisada indeks [36]. Kasutaja uuendamiseks tuleb teha POST päring ja anda kaasa kehatekst (*body*) uuendatud andmetega. Korruga uuendatakse ühe kasutaja andmeid [46]. Olenevalt päringust *Function* kas tagastab andmeid JSON kujul API'na või uuendab SharePointi listis andmeid.

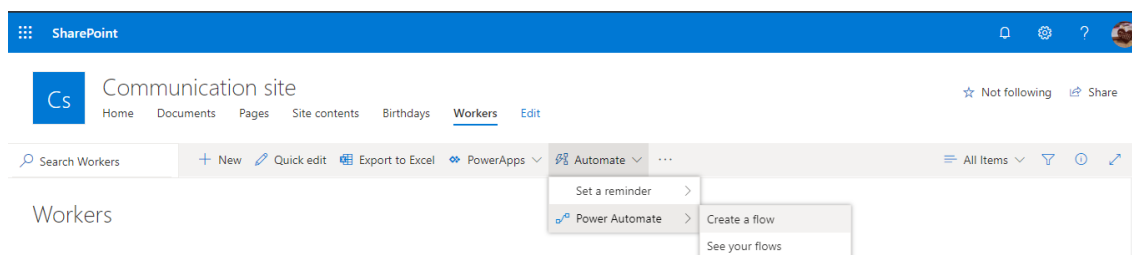
SharePointi listist andmeid pärides ja sinna kirjutades õnnestus peaaegu kõik edukalt, väljaarvatud sellise veeru lugemine ja muutmine, mis on objekt. Selliseks veeruks on

näiteks „LinkedIn“. Nimelt on antud veeru tüüp „hüperlink või pilt“ ja seetõttu on see objekt, millel on omakorda omadused „URL“ ja „kirjeldus“ [46].

### 3.2.3 *Function* andmete sünkroonimiseks

Loodud SharePointi listis saab andmeid muuta mitmel viisil – esiteks läbi *web part*’i ja teiseks otse SharePointi listi muutes. Kui listis andmed muutuvad, siis tahetakse, et andmed muutuksid ka Azure AD’s ja teistes Office’i rakendustes. Selle jaoks loodi Azure *Function* nimega *SyncChangesToAzureADFunc* (Lisa 1 - Programmikood), mis on loodud täpselt sama seadistusega kui eelmine *Function*. *Function* saab sisendiks muudetud listi kirje ning seejärel uuendab kasutaja andmed kasutades *GraphClient*’it ja Microsoft Graph API’t.

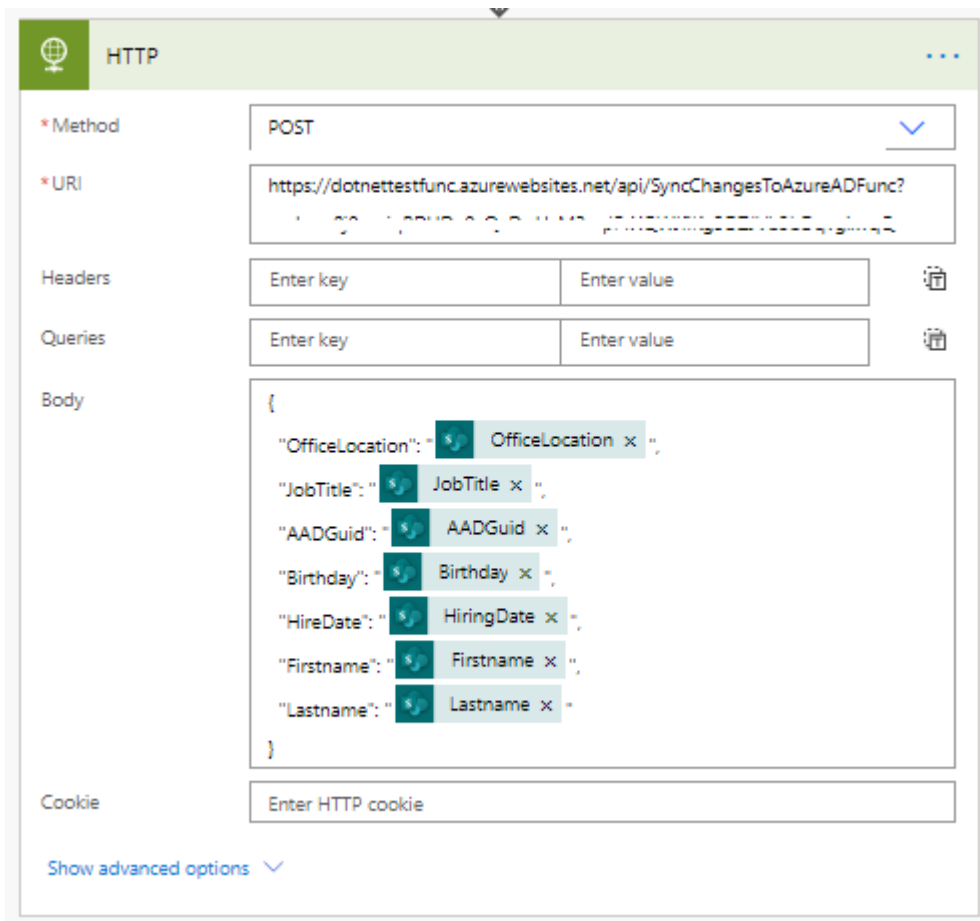
Kuid selleks, et *Function* käivituks, peab miski seda välja kutsuma, tehes päringu *Function*’i URL’iga (*Uniform Resource Locator*), mis saadakse Azure’st *Function*’i juurest. Heaks lahenduseks siinkohal on Microsofti rakendus *Power Automate*, millega saab luua automatiseeritud töövoogusid [47] (Joonis 14).



Joonis 14. *Power Automate*’i töövoole lisamine listile

*Power Automate*’i töövoole saab lisada erinevaid *trigger*’eid ehk käivitajaid ja tegevusi. Loodavas lahenduses lisati töövoole käivitajaks see, kui listi kirjet on muudetud või on listi uus rida lisatud. Ning seejärel alustatakse tegevusega, milleks valiti HTTP, millega on võimalik teha erinevaid HTTP päringuid (Joonis 15).





Joonis 15. Microsoft *Power Automate*'i loomine

Päringu meetodiks valiti POST, sest andmeid tuleb uuendada. URI'ks (*Uniform Resource Identifier*) kopeeriti *Function*'i URL, mis Azure'st saadi (maha on tõmmatud kood). *Show advanced options*'i alt tuleb sisestada ka Azure AD'st saadavad andmed registreeritud rakenduse kohta. Päringu kehasse (*body*) saab anda väärtusi otse SharePointi listist. Ideaalis soovitaks, et kõik listi väljad, mis on olemas ka Microsoft Graph API *User* objekti küljes, uuendataks. AADGuid väärtus antakse *Function*'ile seetõttu, kuna see on ainuke täiesti unikaalne väärtus kasutaja kohta ja selle järgi saab õige kasutaja andmeid uuendada [48], [49].

Kuvatavatest andmetest sai uuendada kõiki andmeid peale meiliaadressi ja AADGuidi, sest need väljad on *read-only* ehk neid saab vaid lugeda ja mitte muuta.

*GraphClient*'iga andmeid uuendades tekkis probleeme kuupäeva formaatidega. Graph API jaoks on tarvis, et kuupäev oleks ISO 8601 formaadis, kuid SharePointi listist tulev kuupäev on teises formaadis [33]. Uuendada ei saanud ka telefoninumbrit ning selle põhjuseks oli see, et loodud rakendusel puudusid vastavad õigused. Microsofti

dokumentatsioonis on kirjas, et kasutajate mobiilinumbrate muutmiseks tuleb rakendusele anda lisaõiguseid rollide näol [33].

### 3.3 *Web part*'i loomine

Luuakse üks *web part*, mis kuvab listi ettevõtte töötajatest, tulevastest sünnipäevadest ja uutest töötajatest. Kui töötaja nimele vajutada, avaneb töötaja profiil. Iga töötaja saab *web part*'is enda profiilil andmeid muuta.

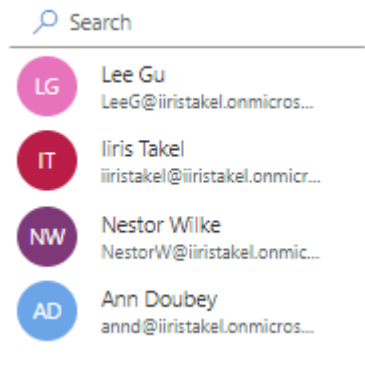
#### 3.3.1 Struktuur

*Web part*'i loomiseks [8] kasutatakse Reacti. Loodava *web part*'i struktuuri jaoks pandi paika Reacti komponendid (Joonis 16).



Joonis 16. *Web part*'i struktuur

Kõige üldisem komponent, mis hõlmab endas kogu lahendust, on *Company* ehk ettevõtte, mis hõlmab endas väiksemaid komponente. Väiksemad komponendid jagati nelja erinevasse kausta, et neid kergem hallata oleks. *WorkersList* kaustas on kolm komponenti, millest *UserList* koosneb kahest komponendist: *SearchBar* ja *UserListSearchResults* ehk otsinguriba ja otsingutulemused. Kui otsinguribale midagi kirjutatakse, siis saadab otsingu komponent kasutajate listi komponendile teavituse koos otsingusõnaga. Seejärel filtreerib kasutajate listi komponent kõikidest kasutajatest välja need, kelle ees-, perekonnanimes või meiliaadressis sisestatud tekst esineb. Otsingutulemuste komponent (Joonis 17) saab sisendiks alati kasutajate listi, mis on filtreeritud või kui otsingusõna on tühi, siis kogu listi, mis *backend* on saatnud (Lisa 1 - Programmikood).



Joonis 17. Kasutajate list otsinguga

Kasutaja profiili jaoks loodi kolm komponenti – üldine komponent *UserProfile* ehk kasutajaprofiil, mis hõlmab endas kahte erinevat komponenti – üks kasutaja profiili vaatamiseks (Joonis 18) ja teine kasutaja andmete muutmiseks (Joonis 19). Kasutaja profiili komponendi sisendiks on profiil, mida avada tahetakse. *State*'i ehk olekuna kasutatakse välju *editUserProfile* ja *isLoggedInUserProfile*, millest esimene määrab, kumba alamkomponenti näidata, kas vaatamise või muutmise oma, teise järgi otsustatakse, kas sisseloginud kasutaja saab avatud profiilil andmeid muuta või mitte. Profiili muutmise komponendis hoitakse olekutena kõiki kasutaja andmeid, mida saab muuta. Esialgu määratakse kõigi väärtuseks sisendina tulnud profiili andmed ning kui midagi muudetakse, siis muudetakse ka olekus väärtus ära [50]. Väljad, mida kasutaja näeb, on järgmised: *birthday* (sünnipäev), *firstname* (eesnimi), *hireDate* (tööle asumise kuupäev), *jobTitle* (ameinimetus), *mail* (meiliaadress), *mobilePhone* (mobiiltelefon), *officeLocation* (kontori asukoht), *lastname* (perekonnanimi), *photo* (pilt), *comment* (kommentaar), *linkedin*, *skype* ja neist saab kasutaja muuta välju: *birthday*, *firstname*, *mobilePhone*, *officeLocation*, *lastname*, *photo*, *comment*, *linkedin*, *skype*.

**IT** Iiris Takel  
Developer

[Edit info](#)

**Contact Information**

<b>Work E-mail</b> iiristakel@iiristakel.onmicrosoft.com	<b>Mobile Number</b> 55527189	<b>Office</b> Tallin
<b>Skype</b> iiristakel	<b>Birthday</b> 15. May 1992	<b>Comment</b> comment
<b>Linkedin</b>	<b>Hire Date</b> 11. April 2020	

Joonis 18. Töötaja enda profiil

**IT** Iiris Takel  
Developer

[Edit info](#)

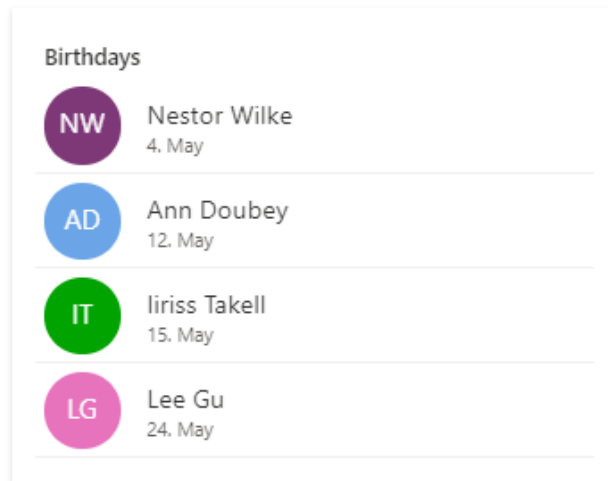
**Contact Information**

<b>Firstname</b> <input type="text" value="Iiris"/>	<b>Birthday</b> <input type="text" value="15/5/92"/>	<b>Job Title</b> <input type="text" value="Developer"/>
<b>Lastname</b> <input type="text" value="Takel"/>	<b>Hire date</b> <input type="text" value="11/4/20"/>	<b>Office location</b> <input type="text" value="Tallin"/>
<b>Work E-mail</b> <input type="text" value="iiristakel@iiristakel.onmicrosoft.com"/>	<b>Skype</b> <input type="text" value="iiristakel"/>	<b>Comment</b> <input type="text" value="comment"/>
<b>Mobile Number</b> <input type="text" value="55527189"/>	<b>Linkedin</b> <input type="text"/>	

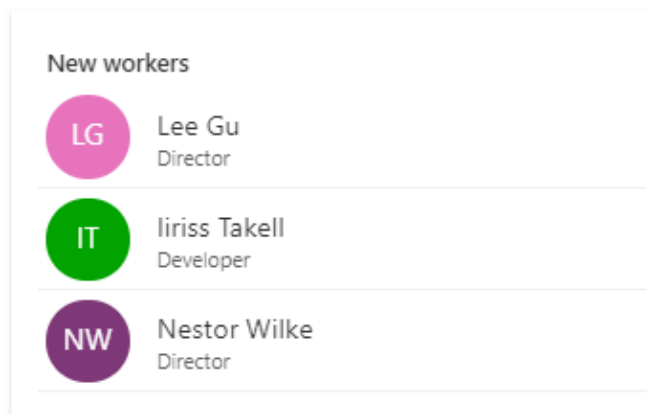
[Save](#)

Joonis 19. Enda andmete muutmise vaade

Kolmas ja neljas kaust sisaldavad sünnipäevade komponenti (Joonis 20) ja uute töötajate komponenti (Joonis 21), mis mõlemad kujutavad endast listi töötajatest, kellel on kas käesoleval kuul sünnipäev või kes on ettevõttesse viimase kahe kuu jooksul tööle tulnud.



Joonis 20. Sünnipäevade *web part*



Joonis 21. Uute töötajate *web part*

### 3.3.2 Andmed

Andmed *web part*'i ja *web part*'ist välja liiguvad läbi loodud *Azure Function*'i. *Azure Function* edastab andmeid JSON-formaadis ning seejärel loetakse need kasutades GET päringut. *Web part*'is kõigi töötajate listi kuvamiseks edastab *Azure Function* SharePointi listist kõik töötajad tööle asumise järjekorras nii, et uued töötajad on listis alati üleval pool. Iga töötaja juures kuvatakse tema ees- ja perekonnanime ning meiliaadressi. Sünnipäevade jaoks on *Function*'il eraldi *endpoint* (URL ressursi saamiseks) „/birthdays“, mis tagastab kuupäevade järgi sorteeritult kõik töötajad, kelle sünnipäev on käesoleval kuul. Andmetest kuvatakse ees- ja perekonnanime ning sünnipäeva. Uute töötajate puhul on *Function*'il *endpoint* „/newWorkers“, mis tagastab tööle asumise järjekorras viimase kahe kuu jooksul lisandunud töötajad. Kuvatakse töötaja nime ja ametipositsiooni.

Kui töötaja enda profiilil andmeid muudab, tehakse POST päring ning andmed edastatakse Azure *Function*'ile, et sealt andmed juba edasi SharePointi listi jõuaksid. Seejärel kuvatakse *web part*'is juba muutunud andmetega profiili ning SharePointi listis muutunud andmed sünkroonitakse teise Azure *Function*'i poolt ka Microsoft Graph API'sse, nii et muutunud andmed on näha ka teistes Office'i rakendustes.

*Web part*'i lisati ka kontroll, et iga töötaja saaks muuta vaid enda profiili andmeid. Selle jaoks tuli SharePointi listi lisada veerg AADGuid ehk Azure's asuv unikaalne tunnus kasutaja kohta. Kontrollimiseks küsiti *GraphClient*'iga sisseloginud kasutaja AADGuid'i väärtust ning võrreldi seda hetkel avatud profiili AADGuid'iga. Kui need väärtused on võrdsed, näeb kasutaja profiili juures muuda (*edit*) nuppu ja saab enda andmeid muuta.

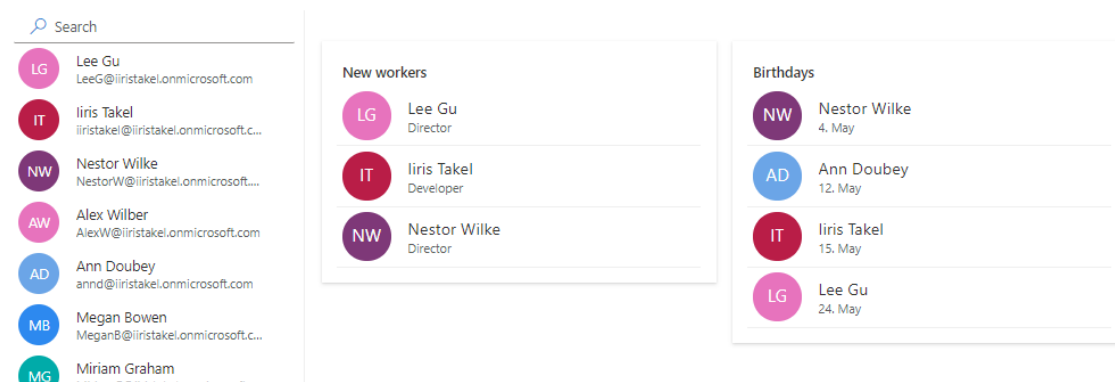
Profiilil andmete kuvamisel tuli vaeva näha kuupäevadega, kuna SharePointi listist tulid kuupäevad formaadis, mida otse välja näidata ei sobinud. Kuid kuupäeva formaati muutes saadi kuupäevast kätte nii päev, kuu kui ka aasta. Kuupäeva muutmiseks kasutati Reacti *DatePicker* komponenti [51].

## 4 Tulemused

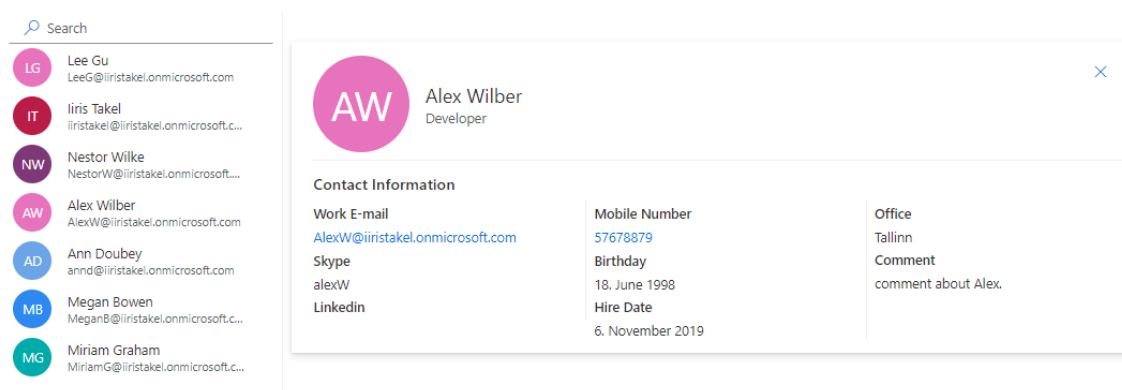
Antud töö käigus loodi SharePointi *web part*, kaks *Azure Function*'it ning SharePointi list ettevõtte töötajatest. Esialgsed nõuded said enamjaolt täidetud, kuid osades nõuetes tuli töö käigus teha muudatusi.

### 4.1 Valminud lahendus

Antud töö käigus valmis SharePointi *web part*, mis kuvab listi organisatsiooni töötajatest, listi tulevatest sünnipäevadest ja uutest töötajatest (Joonis 22). Kui listist on valitud töötaja, kuvatakse valitud töötaja profiili (Joonis 23). Organisatsiooni töötajaid kuvatakse nii, et uuemad töötajad on listis kõrgemal ning samuti on listil olemas otsing, kus saab töötajat otsida üle ees-, perekonnanime ja meiliaadressi.



Joonis 22. *Web part*'i vaade uute töötajate ja sünnipäevadega



Joonis 23. *Web part*'i vaade töökaaslase profiili vaadates

*Web part*'i töötajate listi andmeid päritakse *Azure Function*'iga, kuhu jõuavad andmed SharePointi listist. Kuna SharePointi listis on ka andmeid, mida kõik töötajad näha ei tohi, nagu töötaja töölt lahkumise kuupäev, siis ei antud töötajatele õigust listile ligi pääseda. Listile pääsevad ligi vaid ettevõtte personaliosakonna töötajad, kes saavad listi kaudu

kasutajaid hallata ja nende kohta kõiki välju muuta. Et töötajad aga siiski näeks listis olevat avalikku informatsiooni ja saaks seda muuta, anti listist lugemise ja sinna kirjutamise õigused Azure *Function*'ile, mis edastab avalikud väljad API'na *web part*'ile. Nõue, et personalitöötajad saavad muuta kõikide töötajate andmeid, täideti seega osaliselt, sest nad saavad muuta kõikide kasutajate välju, kuid seda SharePointi listist ja mitte loodud *web part*'ist.

Kui kasutaja avab enda profiili, on profiilil olemas ka muutmise nupp ning kasutaja saab selles samas *web part*'is enda andmeid muuta (Joonis 24). Ka andmete muutmiseks kasutatakse Azure *Function*'it. Muudetud kasutaja profiili andmed edastatakse *Function*'ile ning seejärel muudetakse need ära SharePointi listis.

The screenshot shows a user profile editing interface. On the left, there is a search bar and a list of user avatars with names and email addresses. The main area displays the profile for 'Iiris Takel', a Developer. The profile includes an 'Edit info' button and a 'Contact Information' section with the following fields:

Field	Value
Firstname	Iiris
Lastname	Takel
Work E-mail	iiristakel@iiristakel.onmicrosoft.com
Mobile Number	55527189
Birthday	15/5/92
Hire date	11/4/20
Skype	iiristakel
LinkedIn	
Job Title	Developer
Office location	Tallin
Comment	comment

A blue 'Save' button is located at the bottom left of the form.

Joonis 24. Enda profiili muutmise vaade

Eialgu oli nõuetes kirjas, et *web part*'i töötajate listis tuleb iga töötaja juures kuvada ees- ja perekonnanime, meiliaadressi ning telefoninumbrit, kuid arenduse käigus selgus, et kasutades Office UI Fabric React disainielementidest *Persona* elementi, ei ole võimalik listis kuvada iga *Persona* juures kolme rida teksti ning seetõttu otsustati jääda vaid ees-, perekonnanime ja meiliaadressi kuvamise juurde [13].

Nõuded, et töötajate kontaktandmed on seotud Azure AD kontodega ja et Outlookis kuvatavad andmed oleksid sünkroonis loodud *web part*'ist muudetud andmetega, said täidetud, sest kui töötaja enda andmeid muudab *web part*'is või muudetakse töötaja andmeid SharePointi listist, siis need andmed muutuvad ka mujal Office'i rakendustes, nagu Outlookis. Selle funktsionaalsuse tagamiseks loodi Azure *Function*. *Function*'i käivitab SharePointi listiga seotud töövoog, mis käivitub iga kord, kui listis tehakse muudatus või lisatakse uus rida.



## 4.2 Edasiarendus

Antud tööd on võimalik edasi arendada mitmel viisil. Üheks vajalikuks suunaks on täiustada disaini nii, et lahendus sobiks ka mobiilis mugavalt kasutamiseks. Samuti on võimalik parandada üldist disaini ning kujundust.

Lahendust saaksid kasutada ka teised ettevõtted, kui lahendust täiustada ja paindlikumaks muuta – näiteks teha seadistatavaks, milliseid välju kasutaja profiilil kuvatakse ja milliseid välju kasutaja muuta saab. Samuti saaks teha seadistatavaks selle, kellel on õigus kasutajate profiile otse *web part*'ist muuta.

## Kokkuvõte

Töö eesmärk oli luua ettevõttele, kus kasutatakse Azure AD kontosid, töötajate haldamise keskkond SharePointi. Esimene probleem seisnes selles, et personaliosakond ei saanud muuta töötajate kontodel andmeid, ilma et neile süsteemiadministraatori õiguseid oleks antud. Teine probleem seisnes selles, et polnud keskkonda, kus kõik töötajad näeksid kõiki endaga samas ettevõttes töötavaid inimesi, nende kontaktandmeid, nimekirja uutest töötajatest ja tulevastest sünnipäevadest.

Töö meetodikaks oli välja selgitada ettevõttele vajalikud nõuded, võrrelda juba olemasolevaid Microsofti kasutajate haldamise lahendusi ning tuua välja nende sobivus antud probleemidele. Järgnes tehniline analüüs, kus selgitati välja lahenduse jaoks kasutatavad komponendid ning seejärel realiseeriti lahendus.

Töö käigus loodi ettevõtte personaliosakonnale kõigi töötajate haldamise võimalus, ilma et neile süsteemiadministraatori õiguseid tuleks anda. Kõigi töötajate jaoks loodi aga SharePointi *web part*, kus on näha käesoleva kuu sünnipäevad, uued töötajad ning kõigi töötajate profiilid. Iga töötaja saab muuta enda andmeid selles samas rakenduses ning muudetud andmed sünkroniseeritakse kõikidesse Microsoft Office'i rakendustesse.

*Web part*'i andmevahetuse eest vastutab Azure *Function* ning andmeid hoitakse SharePointi listis. SharePointi listist andmete Azure AD kontodesse sünkroonimisega tegeleb teine Azure *Function*. Personalitöötajatele tuleb lahenduse kasutamiseks anda vaid õigus loodud SharePointi listi lugemiseks ja muutmiseks. Muid õigusi loodud lahenduse kasutamiseks töötajate kontodele lisada pole vaja. Kõik vajaminevad õigused lisati Azure *Function*'itele, mis panevad paika selle, milliseid andmeid töötajad näevad ja milliseid andmeid saab muuta.

Suures ulatuses said töö alguses paika pandud nõuded täidetud. Peamised probleemid tekkisid andmete edastamisega, sest andmete formaadid olid erinevad või ei toetanud Microsoft Graph API tehtavat päringut. Antud tööd arendatakse edasi, et see ettevõttes, mille jaoks lahendus loodi, kasutusele võtta.

Töö autor on loodud töö tulemustega rahul ning õppis töö käigus palju juurde kliendi ja serveri vahelisest suhtlusest.

## Kasutatud kirjandus

- [1] „API ehk rakendusliides,“ [Võrgumaterjal]. Available: <https://help.scoro.ee/manuals/169/?rel=integrations>. [Kasutatud 30 4 2020].
- [2] „What is Azure Active Directory?,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-what-is>. [Kasutatud 10 5 2020].
- [3] „E-teatmik,“ [Võrgumaterjal]. Available: <http://vallaste.ee/>. [Kasutatud 29 4 2020].
- [4] „The Client ID and Secret,“ [Võrgumaterjal]. Available: <https://www.oauth.com/oauth2-servers/client-registration/client-id-secret/>. [Kasutatud 12 5 2020].
- [5] „JSON – Mis on JSON,“ [Võrgumaterjal]. Available: <https://www.metshein.com/unit/json-mis-json/>. [Kasutatud 30 4 2020].
- [6] „Fluent UI,“ [Võrgumaterjal]. Available: <https://developer.microsoft.com/en-us/fluentui#>. [Kasutatud 9 3 2020].
- [7] „Access Tokens,“ [Võrgumaterjal]. Available: <https://auth0.com/docs/tokens/concepts/access-tokens>. [Kasutatud 29 4 2020].
- [8] „Build your first SharePoint client-side web part (Hello World part 1),“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/build-a-hello-world-web-part>. [Kasutatud 20 3 2020].
- [9] P. Laubheimer, „Wireflows: A UX Deliverable for Workflows and Apps,“ 4 12 2016. [Võrgumaterjal]. Available: <https://www.nngroup.com/articles/wireflows/>. [Kasutatud 20 3 2020].
- [10] „What is wireframing?,“ [Võrgumaterjal]. Available: <https://www.experienceux.co.uk/faqs/what-is-wireframing/>. [Kasutatud 29 4 2020].
- [11] Creately, [Võrgumaterjal]. Available: [www.creately.com](http://www.creately.com). [Kasutatud 27 2 2020].
- [12] R. ALLABARTON, „The Definitive Guide: How To Create Your First Wireframe,“ 24 3 2019. [Võrgumaterjal]. Available: <https://careerfoundry.com/en/blog/ux-design/how-to-create-your-first-wireframe/>. [Kasutatud 20 3 2020].
- [13] „Persona,“ [Võrgumaterjal]. Available: <https://developer.microsoft.com/en-us/fluentui#/controls/web/persona>. [Kasutatud 15 3 2020].
- [14] „Set up a Microsoft 365 developer subscription,“ Microsoft, 1 4 2019. [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/office/developer-program/microsoft-365-developer-program-get-started>. [Kasutatud 10 03 2020].
- [15] „Office Delve for admins,“ Microsoft, [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sharepoint/delve-for-office-365-admins>. [Kasutatud 9 3 2020].
- [16] „People,“ [Võrgumaterjal]. Available: <https://outlook.office.com/people>. [Kasutatud 11 3 2020].
- [17] „Delve,“ [Võrgumaterjal]. Available: <https://eur.delve.office.com/>. [Kasutatud 9 3 2020].

- [18] „My account,“ [Võrgumaterjal]. Available: <https://portal.office.com/account/#personalinfo>. [Kasutatud 11 3 2020].
- [19] „Microsoft Azure,“ [Võrgumaterjal]. Available: <https://portal.azure.com>. [Kasutatud 11 3 2020].
- [20] „Manage user profiles in the SharePoint admin center,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sharepoint/manage-user-profiles>. [Kasutatud 11 3 2020].
- [21] „PnP Modern Search Solution,“ [Võrgumaterjal]. Available: <https://github.com/microsoft-search/pnp-modern-search#pnp-modern-search-solution>. [Kasutatud 23 3 2020].
- [22] „React Birthdays Web Part,“ [Võrgumaterjal]. Available: <https://github.com/SharePoint/sp-dev-fx-webparts/tree/master/samples/react-birthdays#react-birthdays-web-part>. [Kasutatud 12 4 2020].
- [23] „Show people profiles on your page with the People web part,“ Microsoft, [Võrgumaterjal]. Available: <https://support.office.com/en-us/article/show-people-profiles-on-your-page-with-the-people-web-part-7e52c5f6-2d72-48fa-a9d3-d2750765fa05>. [Kasutatud 20 4 2020].
- [24] „Samples by JavaScript Framework,“ [Võrgumaterjal]. Available: <https://sharepoint.github.io/sp-dev-fx-webparts/>. [Kasutatud 23 3 2020].
- [25] „Use Office UI Fabric React components in your SharePoint client-side web part,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sharepoint/dev/spfx/web-parts/get-started/use-fabric-react-components>. [Kasutatud 25 3 2020].
- [26] „AngularJS (v1.6.10) directives specific to the Office UI Fabric,“ [Võrgumaterjal]. Available: <https://github.com/ngOfficeUIFabric/ng-officeuifabric#angularjs-v1610-directives-specific-to-the-office-ui-fabric>. [Kasutatud 31 3 2020].
- [27] „Office UI Fabric JS,“ [Võrgumaterjal]. Available: <https://developer.microsoft.com/en-us/fabric-js>. [Kasutatud 31 3 2020].
- [28] „Use the Microsoft Graph API,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/graph/use-the-api>. [Kasutatud 1 4 2020].
- [29] „Complete basic operations using SharePoint REST endpoints,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/complete-basic-operations-using-sharepoint-rest-endpoints>. [Kasutatud 2 4 2020].
- [30] „List users,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/graph/api/user-list?view=graph-rest-1.0&tabs=http>. [Kasutatud 28 2 2020].
- [31] „Use query parameters to customize responses,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/graph/query-parameters>. [Kasutatud 27 2 2020].
- [32] „user resource type,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/graph/api/resources/user?view=graph-rest-1.0>. [Kasutatud 9 3 2020].
- [33] „Update user,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/graph/api/user-update?view=graph-rest-1.0&tabs=http>. [Kasutatud 12 3 2020].
- [34] „SharePoint Search REST API overview,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/sharepoint/dev/general-development/sharepoint-search-rest-api-overview>. [Kasutatud 14 3 2020].

- [35] „Customize permissions for a SharePoint list or library,“ [Võrgumaterjal]. Available: <https://support.office.com/en-us/article/customize-permissions-for-a-sharepoint-list-or-library-02d770f3-59eb-4910-a608-5f84cc297782>. [Kasutatud 15 3 2020].
- [36] „Add an index to a SharePoint column,“ [Võrgumaterjal]. Available: <https://support.microsoft.com/en-gb/office/add-an-index-to-a-sharepoint-column-f3f00554-b7dc-44d1-a2ed-d477eac463b0?ui=en-us&rs=en-gb&ad=gb>. [Kasutatud 5 4 2020].
- [37] „Overview of Microsoft Graph,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/graph/overview#whats-in-microsoft-graph>. [Kasutatud 15 4 2020].
- [38] „Quickstart: Configure a client application to access web APIs,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-configure-app-access-web-apis#add-permissions-to-access-web-apis>. [Kasutatud 20 3 2020].
- [39] D. Geevarghese, „What Azure Functions are and How You Can Use Them to Run Small Applications Effortlessly,“ 16 7 2018. [Võrgumaterjal]. Available: <https://www.cabotsolutions.com/what-azure-functions-are-and-how-you-can-use-them-to-run-small-applications-effortlessly>. [Kasutatud 15 4 2020].
- [40] „Quickstart: Create an Azure Functions project using Visual Studio Code,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-first-function-vs-code?pivots=programming-language-csharp>. [Kasutatud 15 4 2020].
- [41] „BTJ.CSAdvent.AZFunc,“ [Võrgumaterjal]. Available: <https://github.com/BrianTJackett/BTJ.CSAdvent.AZFunc#btjcsadventazfunc>. [Kasutatud 10 4 2020].
- [42] W. Imoh, „Setup Azure Serverless Functions in 6 Simple Steps,“ 19 12 2019. [Võrgumaterjal]. Available: <https://react.christmas/2019/19>. [Kasutatud 14 4 2020].
- [43] „Configure your App Service or Azure Functions app to use Azure AD login,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/azure/app-service/configure-authentication-provider-aad>. [Kasutatud 17 4 2020].
- [44] „Get metadata for a list,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/graph/api/list-get?view=graph-rest-1.0&tabs=csharp>. [Kasutatud 20 4 2020].
- [45] G. K, „Handling CORS Policy In Azure Functions,“ 20 1 2019. [Võrgumaterjal]. Available: <https://www.c-sharpcorner.com/article/handling-cors-in-azure-function/>. [Kasutatud 20 4 2020].
- [46] „Update listItem,“ [Võrgumaterjal]. Available: <https://docs.microsoft.com/en-us/graph/api/listitem-update?view=graph-rest-1.0&tabs=csharp>. [Kasutatud 21 4 2020].
- [47] „Power automate,“ [Võrgumaterjal]. Available: <https://emea.flow.microsoft.com/>. [Kasutatud 4 5 2020].
- [48] L. Ford, „Using Flow with Graph API,“ 4 2 2019. [Võrgumaterjal]. Available: <https://www.lee-ford.co.uk/using-flow-with-graph-api/>. [Kasutatud 21 4 2020].
- [49] F. Appointaire, „[Azure] Call an Azure Function webhook from an automated Flow and a SharePoint list,“ 15 1 2019. [Võrgumaterjal]. Available:

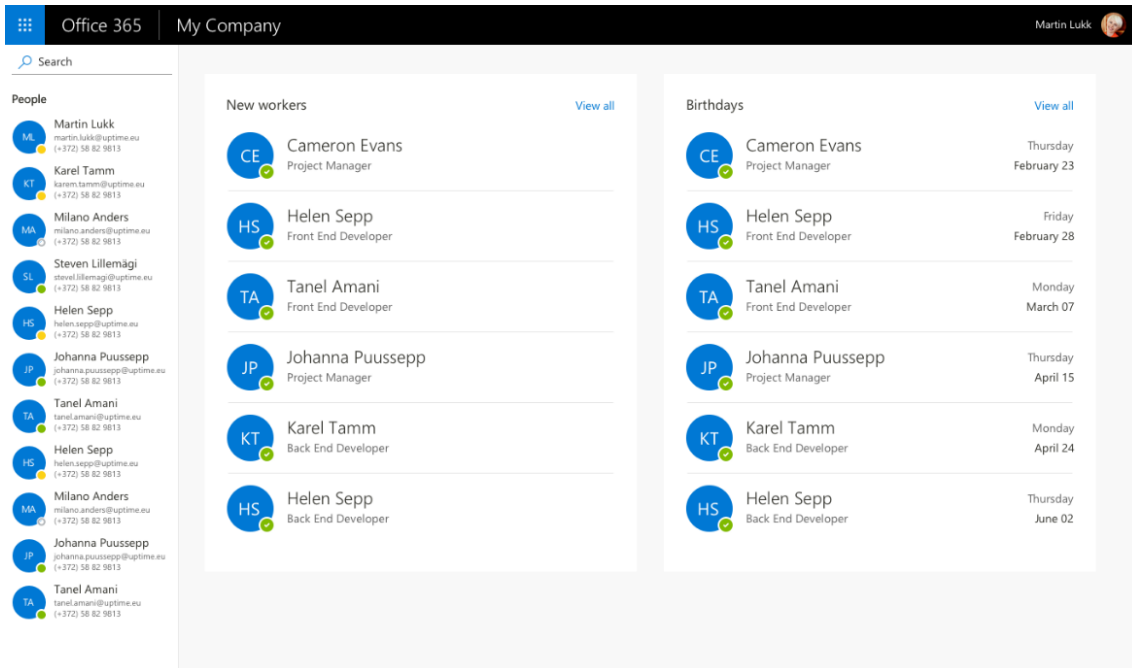
<https://www.starwindsoftware.com/blog/azure-call-an-azure-function-webhook-from-an-automated-flow-and-a-sharepoint-list>. [Kasutatud 21 4 2020].

- [50] „Getting an `<input>` value when form is submitted (React),“ 12 2017. [Võrgumaterjal]. Available: <https://www.freecodecamp.org/forum/t/getting-an-input-value-when-form-is-submitted-react/161870/4> . [Kasutatud 11 4 2020].
- [51] „DatePicker,“ [Võrgumaterjal]. Available: <https://developer.microsoft.com/en-us/fluentui#/controls/web/datepicker>. [Kasutatud 10 4 2020].

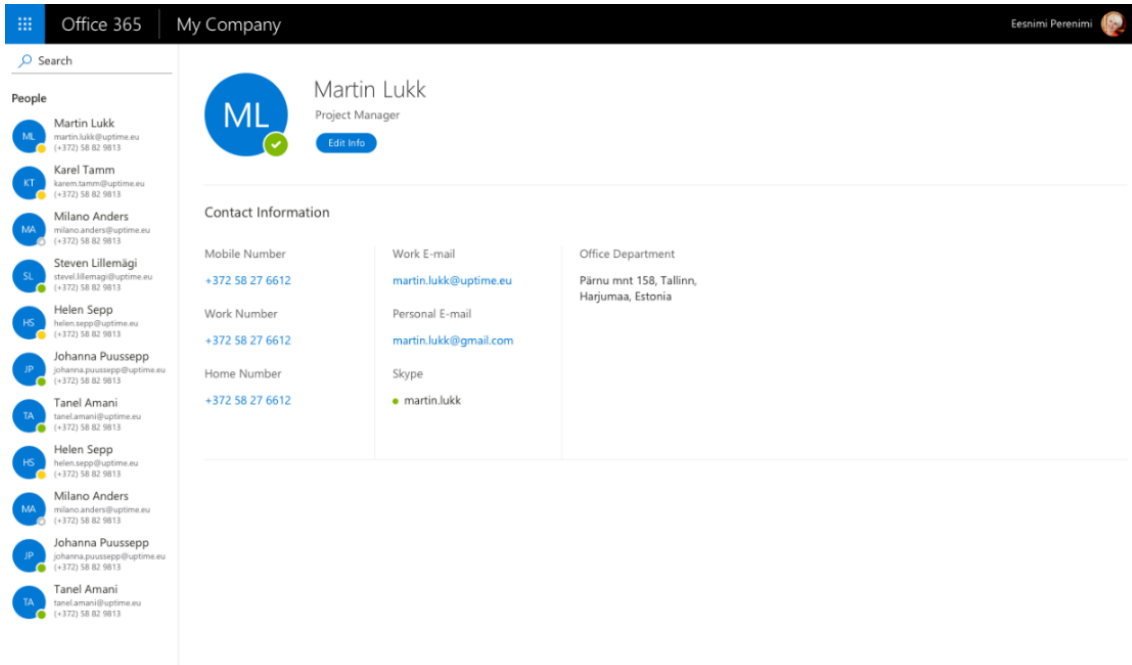
# **Lisa 1 - Programmikood**

Programmikood: <https://github.com/iiristakel/Company>

# Lisa 2 – Desktop vaated UI



Joonis 25. Desktop UI avavaade



Joonis 26. Desktop UI kasutaja profiil vaatamise vaade



- Search
- People
- Martin Lukk  
martin.lukk@uptime.eu  
(+372) 58 82 9813
  - Karel Tamm  
karem.tamen@uptime.eu  
(+372) 58 82 9813
  - Milano Anders  
milano.anders@uptime.eu  
(+372) 58 82 9813
  - Steven Lillemägi  
stevel.lillemagi@uptime.eu  
(+372) 58 82 9813
  - Helen Sepp  
helen.sepp@uptime.eu  
(+372) 58 82 9813
  - Johanna Puissepp  
johanna.puissepp@uptime.eu  
(+372) 58 82 9813
  - Tanel Amani  
tanel.amani@uptime.eu  
(+372) 58 82 9813
  - Helen Sepp  
helen.sepp@uptime.eu  
(+372) 58 82 9813
  - Milano Anders  
milano.anders@uptime.eu  
(+372) 58 82 9813
  - Johanna Puissepp  
johanna.puissepp@uptime.eu  
(+372) 58 82 9813
  - Tanel Amani  
tanel.amani@uptime.eu  
(+372) 58 82 9813

 **Martin Lukk**  
Project Manager  
[Edit Info](#)

Contact Information

<b>Mobile Number</b> <input type="text" value="+372 58 27 6612"/>	<b>Work E-mail</b> <input type="text" value="martin.lukk@uptime.eu"/>	<b>Office Department</b> <input type="text" value="Pärnu mnt 158, Tallinn, Har"/>
<b>Work Number</b> <input type="text" value="+372 58 27 6612"/>	<b>Personal E-mail</b> <input type="text" value="martin.lukk@gmail.com"/>	
<b>Home Number</b> <input type="text" value="+372 58 27 6612"/>	<b>Skype</b> <input type="text" value="martin.lukk"/>	

[Save](#)

Joonis 27. Desktop UI kasutaja profiili muutmise vaade

## Lisa 3 – Graph API'ga kasutaja kohta päritavad andmed

```
{  
  "aboutMe": "string",  
  "accountEnabled": true,  
  "ageGroup": "string",  
  "assignedLicenses": [{"@odata.type": "microsoft.graph.assignedLicense"}],  
  "assignedPlans": [{"@odata.type": "microsoft.graph.assignedPlan"}],  
  "birthday": "String (timestamp)",  
  "businessPhones": ["string"],  
  "city": "string",  
  "companyName": "string",  
  "consentProvidedForMinor": "string",  
  "country": "string",  
  "createdDateTime": "String (timestamp)",  
  "creationType": "string",  
  "department": "string",  
  "displayName": "string",  
  "employeeId": "string",  
  "faxNumber": "string",  
  "givenName": "string",  
  "hireDate": "String (timestamp)",  
  "id": "string (identifier)",  
  "identities": [{"@odata.type": "microsoft.graph.objectIdentity"}],  
  "imAddresses": ["string"],  
  "interests": ["string"],  
  "isResourceAccount": false,  
  "jobTitle": "string",  
  "legalAgeGroupClassification": "string",  
  "licenseAssignmentStates": [{"@odata.type": "microsoft.graph.licenseAssignmentState"}],  
}
```

"lastPasswordChangeDateTime": "String (timestamp)",  
"mail": "string",  
"mailboxSettings": {"@odata.type": "microsoft.graph.mailboxSettings"},  
"mailNickname": "string",  
**"mobilePhone": "string",**  
**"mySite": "string",**  
"officeLocation": "string",  
"onPremisesDistinguishedName": "string",  
"onPremisesDomainName": "string",  
"onPremisesExtensionAttributes": {"@odata.type": "microsoft.graph.onPremisesExtensionAttributes"},  
"onPremisesImmutableId": "string", EI KATSETANUD  
"onPremisesLastSyncDateTime": "String (timestamp)",  
"onPremisesProvisioningErrors": [{"@odata.type": "microsoft.graph.onPremisesProvisioningError"}],  
"onPremisesSamAccountName": "string",  
"onPremisesSecurityIdentifier": "string",  
"onPremisesSyncEnabled": true,  
"onPremisesUserPrincipalName": "string",  
**"otherMails": "string",**  
"passwordPolicies": "string",  
"passwordProfile": {"@odata.type": "microsoft.graph.passwordProfile"},  
**"pastProjects": ["string"],**  
"postalCode": "string",  
"preferredDataLocation": "string",  
**"preferredLanguage": "string",**  
"preferredName": "string",  
"provisionedPlans": [{"@odata.type": "microsoft.graph.provisionedPlan"}],  
"proxyAddresses": ["string"],  
**"responsibilities": ["string"],**  
**"schools": ["string"],**  
"showInAddressList": true,

```

"signInSessionsValidFromDateTime": "String (timestamp)",

"skills": ["string"],

"state": "string",

"streetAddress": "string",

"surname": "string",

"usageLocation": "string",

"userPrincipalName": "string",

"userType": "string",

"calendar": { "@odata.type": "microsoft.graph.calendar" },

"calendarGroups": [ { "@odata.type": "microsoft.graph.calendarGroup" } ],

"calendarView": [ { "@odata.type": "microsoft.graph.event" } ],

"calendars": [ { "@odata.type": "microsoft.graph.calendar" } ],

"contacts": [ { "@odata.type": "microsoft.graph.contact" } ],

"contactFolders": [ { "@odata.type": "microsoft.graph.contactFolder" } ],

"createdObjects": [ { "@odata.type": "microsoft.graph.directoryObject" } ],

"directReports": [ { "@odata.type": "microsoft.graph.directoryObject" } ],

"drive": { "@odata.type": "microsoft.graph.drive" },

"drives": [ { "@odata.type": "microsoft.graph.drive" } ],

"events": [ { "@odata.type": "microsoft.graph.event" } ],

"inferenceClassification": { "@odata.type": "microsoft.graph.inferenceClassification" },

"mailFolders": [ { "@odata.type": "microsoft.graph.mailFolder" } ],

"manager": { "@odata.type": "microsoft.graph.directoryObject" },

"memberOf": [ { "@odata.type": "microsoft.graph.directoryObject" } ],

"messages": [ { "@odata.type": "microsoft.graph.message" } ],

"outlook": { "@odata.type": "microsoft.graph.outlookUser" },

"ownedDevices": [ { "@odata.type": "microsoft.graph.directoryObject" } ],

"ownedObjects": [ { "@odata.type": "microsoft.graph.directoryObject" } ],

"photo": { "@odata.type": "microsoft.graph.profilePhoto" },

"registeredDevices": [ { "@odata.type": "microsoft.graph.directoryObject" } ]
}

```