

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Tair Kokk 185740IADB

Töötajate Azure AD kontode haldus SharePointis

Bakalaureusetöö

Juhendaja: Aleksei Talisainen
Magistrikraad

Tallinn 2021

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Tair Kokk

14.05.2021

Annotatsioon

Konkreetses ettevõttes, kus on kasutusel Office 365, hallatakse töötajate andmeid vanas siseveebis. Töö eesmärgiks on luua lahendus, kus personalitöötajatel oleks võimalik hallata töötajate Azure AD kasutajaid SharePointis selliselt, et neile ei peaks lisama juurde ühtegi Azure AD õigust, sest need annavad üldiselt liiga suured ligipääsud erinevates keskkondades.

Töös kirjeldatakse Office 365 kasutajaprofiile, tutvustatakse nõudeid ja analüüsitakse analoogseid lahendusi ning tehnoloogiaid, millega uus lahendus valmis teha. Lõputöö tulemusena valmis lahendus, mis täidab töös esitatud eesmärgid ja nõudeid. Ettevõtte otsustas autori poolt loodud lahenduse, mida on kavas mõningal määral edasi arendada, kasutusele võtta. Personalitöötajad ei pea enam töötajate andmeid vana siseveebi kaudu haldama.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 26 leheküljel, 5 peatükki, 12 joonist, 0 tabelit.

Abstract

Managing Employees' Azure AD Accounts in SharePoint

In the specific company, where Office 365 is in use, is managing their employees' data on the old intranet. The goal of this thesis is to create a solution where human resource employees could manage employees' Azure AD accounts in SharePoint without having to add any Azure AD permissions to them, as these permissions generally give way too much access.

The author gives an overview of Office 365 user profiles, introduces the requirements and analyzes similar solutions and technologies to create a solution with. The solution which was created fulfilled the goal and met the requirements. The company decided to implement the solution which was created by the author. There are also some plans to continue the development of the application. Human resource employees don't have to manage employees' data on old intranet anymore.

The thesis is in Estonian and contains 26 pages of text, 5 chapters, 12 figures, 0 tables.

Lühendite ja mõistete sõnastik

.NET Core	Vabavaraline platvorm erinevate rakenduste ehitamiseks.
API	<i>Application Programming Interface</i> . Rakendusliides.
Azure	Microsofti pilveteenuste platvorm.
Azure AD	<i>Azure Active Directory</i> . Microsofti pilvepõhine identiteedi ja juurdepääsu haldamise teenus.
AWS	<i>Amazon Web Services</i> . Amazoni pilveteenuste platvorm.
C#	Programmeerimiskeel.
Exchange	Microsofti meiliserver.
GB	<i>Gigabyte</i> . Ühik.
HTTP	<i>Hypertext Transfer Protocol</i> . Võrguprotokoll.
JSON	<i>JavaScript Object Notation</i> . Andmeformaad.
NuGet	Tööriist pakside halduseks .NET'is.
SDK	<i>Software Development Kit</i> . Tarkvaraarenduse komplekt.
SPFx	<i>SharePoint Framework</i> . Raamistik SharePointi ja Teamsi lahenduste loomiseks.
SQL	<i>Structured Query Language</i> . Päringukeel andmebaasidega suhtlemiseks.

Sisukord

1 Sissejuhatus	8
2 Analüüs.....	9
2.1 Nõuded.....	9
2.2 Office 365 rakenduste kasutajaprofiilid	9
2.3 Testkeskkond	10
2.4 Analoogsed lahendused	11
2.4.1 Azure AD.....	11
2.4.2 Microsoft 365 administraatorikeskus	12
2.4.3 Exchange'i halduskeskus.....	13
2.4.4 SharePointi kasutajaprofiilid	14
2.4.5 Teise tudengi lõputööna loodud lahendus	16
2.4.6 Analoogsete lahenduste kokkuvõte	17
2.5 Tagarakendus.....	17
2.6 Esirakendus.....	19
3 Teostus.....	20
3.1 SharePointi sait ja loend	20
3.2 Tagarakendus.....	21
3.2.1 Projekti tooriku loomine ja Azure'i seadistamine	21
3.2.2 Microsoft 365 integratsioon	22
3.2.3 Masspäringud	24
3.2.4 Azure AD kontode sünkroniseerimine SharePointi loendisse.....	24
3.2.5 Azure AD kontode uuendamine SharePointi loendis olevate kirjete põhjal .	26
4 Tulemused ja edasiarendus	28
5 Kokkuvõte	30
Kasutatud kirjandus	31
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	33
Lisa 2 - Rakenduse lähtekood.....	34

Jooniste loetelu

Joonis 1. Azure AD konto andmete sünkroniseerimine Office 365 rakenduste kasutajaprofiilidesse.	10
Joonis 2. Azure AD kasutajate loend Azure'i portaalis.	11
Joonis 3. Azure AD kasutaja muutmise vaade.	12
Joonis 4. Microsoft 365 administraatorikeskuse kasutajate vaade.	12
Joonis 5. Microsoft 365 administraatorikeskuse kasutaja muutmise vaade.	13
Joonis 6. Exchange'i halduskeskuse postkastide vaade.	13
Joonis 7. Exchange'i kasutaja muutmise vaade.	14
Joonis 8. SharePointi kasutajate loend.	14
Joonis 9. SharePointi kasutaja muutmise vaade.	15
Joonis 10. Riigi veeru vormindus JSON kujul.	21
Joonis 11. Kasutajate pärimine kasutades Microsoft Graph SDK'd.	23
Joonis 12. Microsoft Graph API'ga andmete pärimine.	23

1 Sissejuhatus

Konkreetses ettevõttes on kasutusel Office 365 ning seal kasutatakse Azure AD(*Active Directory*) kontosid. Hetkel haldavad personalitöötajad töötajate andmeid vanas siseveebis, kuigi vajalikud väljad on Azure AD kontol olemas. Selle põhjuseks on see, et teiste Azure AD kontode haldamiseks on vaja vastavaid Azure AD õiguseid, kuid need annavad liiga paljudele andmetele ligipääsu. Lisaks on probleemiks see, et töötajate kontosid saab vaikimisi hallata keskkondades, mis on mõeldud administraatoritele ning personalitöötajatel ei tohiks nendele ligipääsu olla.

Töö eesmärgiks on luua lahendus, kus personalitöötajatel on võimalik hallata töötajate Azure AD kontosid selliselt, et haldajatele poleks vaja lisada juurde Azure AD õiguseid. Selle tulemusena puuduks vajadus kasutada vanat siseveebi töötajate andmete haldamiseks.

Käesolev töö koosneb kolmest osast. Esimeses osas pannakse paika nõuded, mida lahendus täitma peab, tutvustatakse Office 365 kasutajaprofiile ning analüüsitakse analoogseid lahendusi ning esi- ja tagarakendust. Teises osas antakse ülevaade praktilisest tööst, kus selgitatakse esinenud probleeme ning neile leitud lahendusi. Kolmas osa, mis on mahult väiksem, keskendub tehtud töö tulemustele ning võimalikule edasiarendusele.

2 Analüüs

2.1 Nõuded

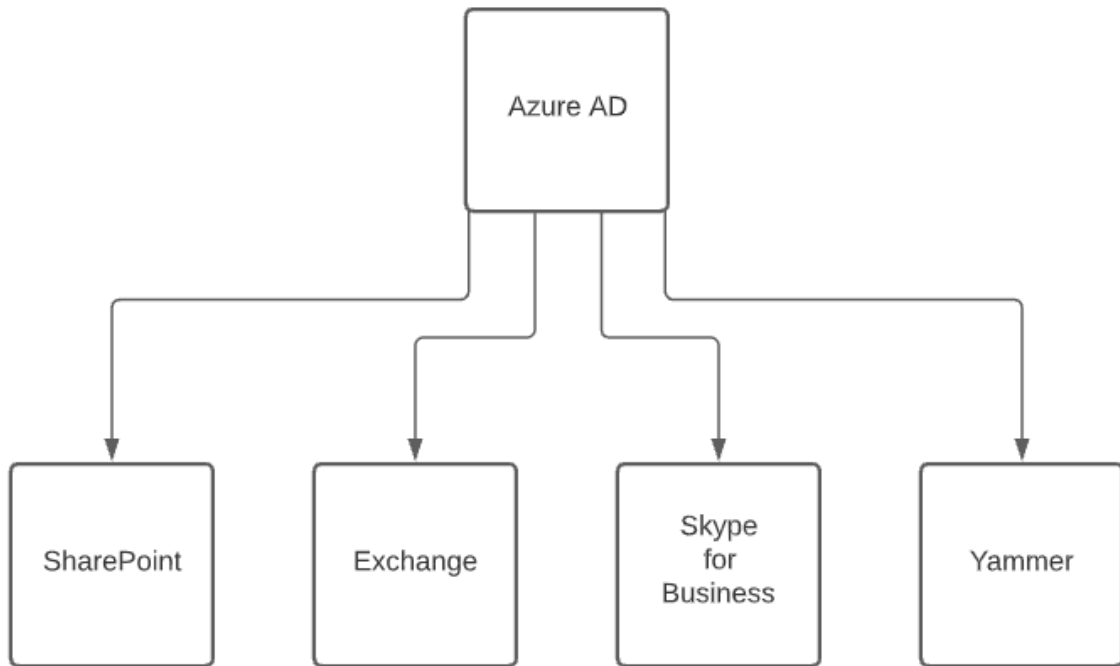
Nõuete väljaselgitamiseks arutleti koos konkreetse ettevõtte personalitöötajatega, kes peamiselt loodavat lahendust kasutama hakkavad, ning tehnoloogiajuhiga. Jõuti järeldusele, et lahendus, mis luuakse, peab vastama järgmistele nõuetele:

- personalitöötajad peavad saama muuta töötajate Azure AD kontode järgmisi väärtuseid: profiilipilt, kuvatav nimi, eesnimi, perekonnanimi, töötelefon, ametikoht, sünnikuupäev, tööle asumise kuupäev, kontori asukoht ja riik,
- Azure AD kontosid peab saama hallata SharePointis,
- teisi väärtuseid, mida eelnevalt ei nimetatud, ei tohi saada muuta,
- peab saama hallata, et kellel on lahendusele ligipääs,
- teiste Azure AD kontode andmeid ei tohi olla näha ning neid ei tohi saada hallata,
- töötajaid peab saama otsida nime järgi,
- töötajaid peab saama sorteerida nime järgi
- personalitöötajatele ei tohi juurde lisada ühtegi Azure AD õigust,
- töötajatel peab olema võimalus näha teiste töötajate telefoninumbreid,
- peab saama hallata, et milliseid välju on näha loendis ning andmete muutmise vormis.

2.2 Office 365 rakenduste kasutajaprofiilid

Office 365 koosneb mitmetest rakendustest, näiteks SharePoint, Exchange ja Skype for Business, ning mõnedel nendel on omad kasutajaprofiilid, mida hoiustatakse erinevatel

viisidel, näiteks SharePointil on selleks enda SQL(*Structured Query Language*) andmebaas. Selle põhjuseks on see, et need rakendused olid kunagi iseseisvad lahendused. Nende põhiallikaks on Azure AD kasutaja, mille mõningate väljade väärtused sünkroniseeritakse teistesse Office 365 rakenduste kasutajaprofiilidesse (Joonis 1). Selleks ei pea midagi seadistama, vaid kõik toimub automaatselt [1].



Joonis 1. Azure AD konto andmete sünkroniseerimine Office 365 rakenduste kasutajaprofiilidesse.

Seega on Azure AD konto andmed nähtaval erinevatest kohtadest ning nende haldamiseks on erinevaid võimalusi.

2.3 Testkeskkond

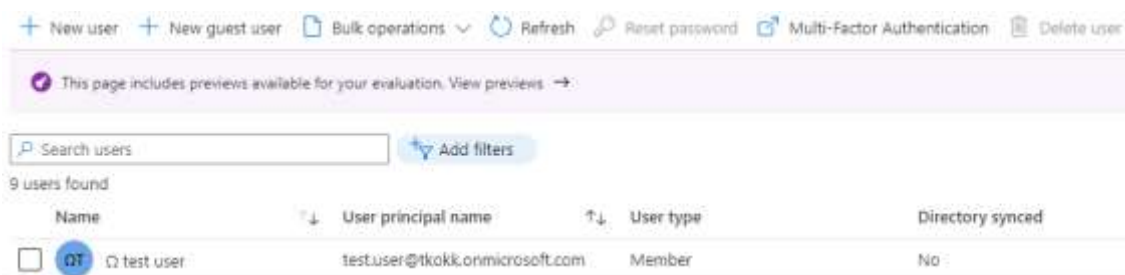
Kuna lahendus on seotud Office 365ga, siis on peamiselt kaks võimalust: kasutada autori enda keskkonda või ettevõtte testkeskkonda. Kindlasti saaks kasutada ka ettevõtte päris keskkonda, aga see ei oleks mõistlik, kuna vigade, mida oleks saanud vältida, tekkimisel võivad olla halvad tagajärjed ning lisaks ei ole autoril seal kõiki õigusi, mis tähendab seda, et ei oleks võimalik teha kõiki tegevusi iseseisvalt. Kuna konkreetne ettevõtte väga suuri arendusi enda Office 365te ei plaani, siis ei ole mõtet luua sinna testkeskkonda. Seega otsustati autori enda Office 365 kasuks, mis oli tal ennem juba olemas. Selleks oli vaja liituda Microsoft 365 arendaja programmiga. Alguses on see 90 päeva tasuta ning kui sellega tegeletakse aktiivselt, siis Microsoft uuendab tellimust automaatselt ja täiesti tasuta [2].

2.4 Analooesed lahendused


Uuritakse viite lahendust, nendest neli on loodud Microsofti poolt ning neid ei pea eraldi integreerima, vaid on juba vaikimisi olemas. Viimane lahendus on loodud endise tudengi poolt. Olemasolevate lahenduste testimiseks loodi uus Azure AD kasutaja administraatori õigusteta ning litsensita. Ennem iga lahenduse testimist antakse kontole ainult tavakasutaja õigused.

2.4.1 Azure AD

Azure AD'sse, kus on võimalik hallata ka väga palju teisi seadeid ja objekte, peale kasutajate, pääseb Azure'i portaali kaudu. Kasutajate loendis (Joonis 2) saab otsida ning filtreerida kontosid erinevate parameetrite põhjal. Lisaks on võimalik reguleerida, kuid mitte väga suurel määral, milliseid välju kuvatakse. Näiteks ei ole võimalust välja näidata töötelefoni. Käesolevas lahenduses on ülemäära palju andmeid, kuid siiski puuduvad nõutavatest väljadest sünnipäev ning tööle tuleku kuupäev.



The screenshot shows the Azure AD user management interface. At the top, there are several action buttons: '+ New user', '+ New guest user', 'Bulk operations', 'Refresh', 'Reset password', 'Multi-Factor Authentication', and 'Delete user'. Below these is a notification bar stating 'This page includes previews available for your evaluation. View previews'. A search bar contains 'Search users' and an 'Add filters' button. Below the search bar, it says '9 users found'. The main part of the screenshot is a table with the following columns: 'Name', 'User principal name', 'User type', and 'Directory synced'. The first row shows a user with a blue profile picture, the name 'test user', the user principal name 'test.user@tkokk.onmicrosoft.com', the user type 'Member', and the directory synced status 'No'.

Name	User principal name	User type	Directory synced
 test user	test.user@tkokk.onmicrosoft.com	Member	No

Joonis 2. Azure AD kasutajate loend Azure'i portaalis.

Ilma õigusteta kasutaja saab vaadata teiste kontode andmeid ning muuta ainult enda kasutajat. Seetõttu lisati testkontole kasutajaadministraatori roll, millega kaasnesid piisavad õigused. Sellega tekkis uus probleem, sai näha ja muuta liiga palju objekte, näiteks oli võimalik hallata gruppe ning kõiki kontosid. Nii palju õiguseid ei tohiks personalitöötajatel kindlasti olla. Lisaks sai redigeerida andmeid, mida nõuetes kirjas ei olnud, näiteks joonisel (Joonis 3) on näha, et on olemas *Manager* väli.

The screenshot shows the Azure AD user profile for 'test user' (test.user@tkokk.onmicrosoft.com). The profile includes a profile picture with the letters 'QT' and a 'User Sign-ins' chart showing activity from March 18 to April 11. The 'Identity' section lists the user's name, principal name, object ID, and source (Azure Active Directory). The 'Job info' section shows the user's title as 'tester', department as 'testing', and manager.

Joonis 3. Azure AD kasutaja muutmise vaade.

2.4.2 Microsoft 365 administraatorikeskus

Microsoft 365 administraatorikeskus on mõeldud administraatoritele ja tavakasutajatel sinna ligipääs puudub. See on üldine administreerimiskeskus. Nagu eelneva lahenduse puhul, lisati kontole kasutajaadministraatori roll. Kasutajate vaates (Joonis 4) on peamised väljad olemas. Kuvatavaid välju on võimalik muuta ainult teatud määral. Samuti on olemas nime põhjal sorteerimine ning otsimine.

Aktiivsed kasutajad

The screenshot shows the 'Aktiivsed kasutajad' (Active Users) view in the Microsoft 365 administrator interface. It features a toolbar with actions like 'Lisa kasutaja', 'Kasutajamallid', 'Lisa mitu kasutajat', 'Mittikautentimine', 'Kustuta kasutaja', 'Värskenda', 'Lähtesta parool', and 'Elisporid kasutajad'. Below the toolbar is a table with columns for 'Kuvatav nimi', 'Kasutajanimi', 'Ühendid', and 'Väljaveerud'. The first row shows the user 'test user' with the email 'test.user@tkokk.onmicrosoft.com' and the role 'Microsoft 365 55 Developer (ilma Windowsi ja kõnekoiget)'.

Joonis 4. Microsoft 365 administraatorikeskuse kasutajate vaade.

Kasutaja redigeerimise vormis (Joonis 5) on liiga palju andmeid, kuid samas ei ole võimalik muuta sünnipäeva ning tööle tuleku kuupäeva, ülejäänud nõutud väljadest on olemas. Lisaks ei tohiks personalitöötajatel olla ligipääsu administraatorikeskusele.

Halda kontakte

Eesnimi
test

Perekonnanimi
test

Kuvatav nimi *
D test user

Ametinimetin
test

Osakond
test

Office

Töötelefon
Faksinumber

Mobiiltelefon

Tähtsaja

Salvesta muutused

Joonis 5. Microsoft 365 administraatorikeskuse kasutaja muutmise vaade.

2.4.3 Exchange'i halduskeskus

Exchange'i halduskeskus on mõeldud Exchange Online teenuse haldamiseks [3]. Seal on 2 kasutajaliidese versiooni: moderne ning klassikaline. Nende vahel saab väga lihtsalt vahetada, vajutades ainult ühte nuppu. Otsustati kasutada uuemat versiooni, kuna see on kasutajasõbralikum ja sarnane teiste halduskeskustega ning võib eeldada, et kunagi tulevikus klassikalist vaadet ei eksisteeri.

Et kasutaja Exchange'i halduskeskusesse ligi saaks, määrati talle Exchange'i administraatori roll. Kuna siin hallatakse postkaste, siis lisati kasutajale ka vastav litsents, muidu teda postkastide vaates (Joonis 6) näha ei oleks. Postkaste on võimalik otsida ja sorteerida erinevate parameetrite järgi. Kuvatavaid välju on võimalik teatud määral reguleerida, kuid pildi vaatamiseks peab iga postkasti peale eraldi vajutama, mis on üpriski tüütu.

Mailboxes

Add a shared mailbox
 Set default message size restrictions
 Refresh

Display name	Email address	Recipient type	Active status	Last modified time	Options column
D test user	test@domain.microsoft.com	UserMailbox	True	2/14/2017 3:00 PM	

Joonis 6. Exchange'i halduskeskuse postkastide vaade.

Redigeerimise vaates (Joonis 7) on liiga palju välju ning ei ole võimalik hallata sünnipäeva ning tööle tuleku kuupäeva. Kasutaja profiilipilti on võimalik küll vaadata, kuid mitte muuta. Endiselt jääb suuremaks probleemiks see, et personalitöötajatele tuleks anda märkimisväärsed õigused.

Edit contact information

First name:
test

Last name:
user

Display name:
test user

Hide from address lists

Contact information

Job title:
tester

Department:
testing

Office:

Office phone:

Mobile phone:

Home phone:

Fax number:

Street address:

Save Cancel

Joonis 7. Exchange'i kasutaja muutmise vaade.

2.4.4 SharePointi kasutajaprofiilid

SharePointi kasutajaprofiilidesse pääseb läbi SharePointi administreerimiskeskuse, kuid sinna saamiseks on vaja vastavaid õigusi. Selleks lisati testkasutajale SharePointi administraatori roll. Kasutajaprofiilide avalehel (Joonis 8) on järgnevad miinused: vaikimisi on loend tühi, ei ole võimalik muuta kuvatavaid välju, otsing on aeglane ning näeb kõiki kontosid, mitte ainult töötajate omi.

User Profiles

Use this page to manage the user profiles of the User Profile Service application. From this page you can also manage a user's personal site.

Total number of profiles: 11

Find profiles: test Find

Manage Sub-items: Select a sub-type to filter the list of profiles: Default User Profile Subtype















Account name	Display name	E-mail address
test@msn.com	test	test@msn.com
test@msn.com	test	test@msn.com
test@msn.com	test	test@msn.com

Joonis 8. SharePointi kasutajate loend.

Kasutaja redigeerimise vaates (Joonis 9) on liiga palju andmeid ja seadeid, mida on võimalik muuta. Nõutavatest väljadest on puudu ainult riik. Azure AD kontrol on kaks tööle tuleku välja: *hireDate* ja *employeeHireDate* [4]. Esimene väli on SharePointi spetsiifiline ning SharePointi kasutajaprofiilid kasutab seda. Seega oleks teine väli parem lahendus, kuna see on üldisem ning seda soovitab ka Microsoft [4]. Põhiprobleem antud lahenduses on see, et enamus nõutud välju ei salvestata Azure AD kontole, vaid need jäävad ainult SharePointi kasutajaprofiilile. Kui Azure AD konto välju muuta, siis mingi aja tagant sünkroniseeritakse tehtud muudatused SharePointi kasutajaprofiilidesse, mis tähendab seda, et need väljad kirjutatakse üle ning neid ei ole mõtet muuta SharePointi kaudu.

Kasutajaprofiilid

Sellel lehel saate redigeerida seda kasutajaprofiili, muutes järgmiste atribuutide väärtusi. Välise andmeallikaga vastendatud atribuudid kirjutat:

		Kuva
 Kontonimi:	<input type="text" value="i:0#.f membership test.user@tkokk.onmicrosoft.com"/>	Kõik
 Eesnimi:	<input type="text" value="test"/>	Kõik
 Perekonnanimi:	<input type="text" value="user"/>	Kõik
 Nimi: *	<input type="text" value="Ω test user"/>	Kõik
 Töötelefon: *	<input type="text"/>	Kõik
 Osakond: *	<input type="text" value="testing"/>	Kõik
 Pealkiri: *	<input type="text" value="tester"/>	Kõik
Osakond:	<input type="text" value="testing"/> 	Kõik ▾
 Haldur:	<input type="text"/>  	Kõik
Minu teave:	<input type="text"/>	Kõik
Sisestage isiklik kirjeldus, mis väljendaks seda, mida soovite, et teised teie kohta teaksid.		
Isiklik sait:	<input type="text" value="http://"/>	Kõik
Pilt:	 <input type="button" value="Laadi pilt üles"/> Saate laadida üles pildi, mis aitaks teistel teid koosolekutel ja üritustel hõlpsasti ära tunda.	Kõik
 Veebisait:	<input type="text" value="http://"/>	Kõik
 Avaliku saidi ümbersuunamine:	<input type="text" value="http://"/>	Kõik

Joonis 9. SharePointi kasutaja muutmise vaade.

2.4.5 Teise tudengi lõputööna loodud lahendus

Lõputöö „Azure AD’ga integreeritud keskkonna loomine töötajate kontaktide haldamiseks SharePointis“ on loodud 2020. aastal. Selle käigus loodi lahendus, kus personalitöötajad saavad hallata töötajate kontaktandmeid ning lisaks saab iga töötaja muuta enda andmeid. Valminud lahendus koosnes järgmistest osadest:

- SharePointi loend töötajatest,
- veebiossa, kus iga töötaja saab muuta enda andmeid ning personalitöötajad kõikide teiste andmeid veel lisaks. See veebiossa kuvab ka kõiki töötajaid ja ettetulevaid sünnipäevi,
- Azure Functionsi funktsioon, mis tagastab vastavad töötajate andmed, kui veebiossa sinna päringu teeb ning selle kaudu uuendatakse ka SharePointi loendis olevaid kirjeid,
- Azure Functionsi funktsioon, mis sünkroniseerib loendi andmed Azure AD kontodele,
- Power Automate’i HTTP(*Hypertext Transfer Protocol*) käivituse, mis saadab HTTP päringu eelnevas punktis kirjeldatud Azure Functionsi, kui SharePointi loendi kirjet on muudetud [5].

Peamine erinevus teise tudengi lahendusega võrreldes on see, et seal saavad lisaks personalitöötajatele enda andmeid muuta ka töötajad ise, käesolevas töös sellist nõuet ei ole, kuna leiti, et iga töötaja saab piisaval määral enda andmeid muuta vaikimisi olemasolevate lahendustega.

Veebiossa kuvab töötajad ühes kitsas tulbas ning seetõttu kuvatakse algselt ainult täisnimi ja meiliaadress, et näha kõiki andmeid, peab kasutaja peale vajutama. Seega, kui tahta muuta või ümber paigutada kuvatavaid välju, peab täiendama veebiossa programmikoodi. Loodud lahenduses on veebiossa loomine põhjendatud sellega, et töötajad ei tohi kõiki välju näha, samas personalitöötajad peavad neid nägema ning selliselt välju filtreerida SharePointi loendi vaates ei ole võimalik [5]. Veel üheks veebiossa miinuseks on see, et töötajaid ei saa sorteerida nime järgi, samas otsimine on võimalik. Kui tahta kasutada

sama veebiosa käesolevas lahenduses, siis tuleks see nii suurel määral ümber teha, et lihtsam oleks luua uus.

Lahenduses kasutatud Power Automate'i HTTP päringu käivitus muudeti tasuliseks 01.02.2019 [6]. Kuna ettevõtte, kellele lahendus plaanitakse luua, kuskil mujal vastavat litsentsi ei vaja, siis ei ole mõtet ühe väikese tegevuse tõttu seda soetada. Vastava litsentsi ostmist võiks kaaluda ka siis, kui teisi lahenduskäike ei oleks. Selle lahenduse puhul oleks HTTP käivitus võimalik asendada Graph API(*Application Programming Interface*) muudatuse teadetega. See lisaks keerukust, sest ei piisa ainult selle loomisest ühe korra, vaid seda tuleb uuendada iga teatud aja tagant, kuna muudatuse teated aeguvad. Lisaks see annab ainult teada, et vastavas loendis muutus midagi, et teada saada, mis muutus, tuleks hoida kuskil väärtust, millal viimati funktsioon käivitus ning seda võrrelda SharePointi loendi kirjade viimati muudetud välja väärtusega. [7].

Teise tudengi lahenduses tuleks teha nii palju muudatusi, et lihtsam oleks uus lahendus programmeerida, sellepärast ei ole mõistlik seda kasutusele võtta.

2.4.6 Analoogete lahenduste kokkuvõte

Analüüsiti mitmeid erinevaid lahendusi, kuid nendest ei sobinud mitte ükski. Microsofti poolt pakutud lahendustel olid peamiselt järgmised probleemid:

- ei asunud SharePointi keskkonnas,
- personalitöötajatele pidi juurde lisama Azure AD õiguseid,
- mõni nõutud väljadest oli alati puudu,
- kuvatavaid välju sai muuta ainult teatud määral,
- muutmise vormil ei olnud võimalik välju muuta.

Teise tudengi lahenduses oleks vaja teha liiga palju muudatusi.

2.5 Tagarakendus

Loodav rakendus on iseseisev ning ei ole mõne suurema projekti üks osa. Tagarakenduses otsustati kasutada C# programmeerimiskeelt, kuna autoril oli sellega kõige rohkem

kogemust ning teiste programmeerimiskeelte ja eriti nendega seonduvate tehnoloogiate õppimine oleks võtnud liigselt aega.

Lisaks otsustati, et kasutatakse *serverless* tehnoloogiat järgmistel põhjustel:

- servereid ei pea ise haldama, nende eest vastutab teenusepakkuja,
- kulud tulevad väiksemad, kuna loodav rakendus ei tööta ööpäev läbi ning on võimalus maksta täpselt nii palju, kui palju ressursse rakendus kasutab ehk kui see ei tööta, siis selle eest arve ei suurene,
- automaatne skaleeritavus [8].

Viimane punkt ei ole küll käesoleva lahenduse puhul väga oluline, aga see on üks väga positiivne omadus. Ühed populaarsemad *serverless* lahendused on Azure Functions ning AWS(*Amazon Web Services*) Lambda.

Azure Functions on Microsofti üks *serverless* lahendustest. See koosneb funktsioonidest, millel on erinevaid tüüpi käivitusi [9]. Käesolevaks tööks on vaja HTTP päringu ja taimeri käivitusi, mis on mõlemad Azure Functionsil olemas, samuti saab seal kasutada C#'i. Rakenduse pilve viimiseks on vaja Azure'i kontot koos tellimusega, lokaalselt testides seda vaja ei ole. Siis peab valima plaani, mille järgi arvutatakse kulud. Kuna loodaval rakendusel jõudlus ei oma väga tähtsust, siis on mõtekam valida kõige lihtsam, milleks on tarbimise plaan. Sellega saab iga kuu teha 1 000 000 päringut ja kasutada 400 000 GB arvutamise mahtu tasuta ning vaikimisi on ühe funktsiooni ajapiiranguks 5 minutit, aga seda on võimalik muuta 10 minuti peale, mis on loodavale rakendusele täiesti piisav [10], [11]. Igale Azure Functionsile on vaja teha hoiustamise konto, kuhu salvestatakse näiteks konfiguratsiooni andmed.

AWS Lambda põhineb samuti funktsioonidel, mida saab erinevatel viisidel käivitada [12]. Käesolevaks tööks vajaminevad HTTP päringu ja taimeri käivitused on olemas, lisaks saab kasutada programmeerimiskeelt C# [13], [14]. Tasuta kasutamine on täpselt sama, mis Azure Functionsil, ehk 1 000 000 päringut ja 400 000 GB(*Gigabyte*) arvutamise mahtu iga kuu [15]. Funktsiooni ajapiiranguks on 15 minutit, mis on 5 minutit rohkem, kui eelnevalt kirjeldatud lahendusel [16].

Lahendust on võimalik luua mõlema variandiga, aga kuna konkreetses ettevõttes on kasutusel Office 365, kus nagunii tuleb Azure'i kasutada, siis otsustati valida Azure Functions, et kõik oleks ühes keskkonnas olemas.

2.6 Esirakendus

Käesoleva töö nõuetes on kirjas, et rakendus peab olema SharePointis, seega on esirakenduse puhul 2 varianti: luua veebiosa või kasutada olemasolevat SharePointi loendivaadet.

SharePointi loendivaates saab kirjeid kuvada, lisada, muuta, otsida, filtreerida ja sorteerida, luua erinevaid vaateid ja valideerida välju, sellel on veel suurel hulgal funktsionaalsusi. Loendivaadet on võimalik veel täiendada kasutades SPFx'i (*SharePoint Framework*) laiendusi. Veebiosa kõige suurem eelis on see, et see on täielikult erilahendus ehk seal saab teha täpselt nii nagu vaja, kuigi see tähendab seda, et kõik funktsionaalsused tuleb ise luua. Otsustati SharePointi loendivaate kasuks, kuna suurel määral on seal vajalikud funktsionaalsused olemas ning mida ei ole, need on võimalik luua, kasutades SPFx'i laiendusi.

3 Teostus

Selles peatükis kirjeldatakse terve rakenduse, mille autor tegi, loomist, sealhulgas esinenud probleeme ning nendele leitud lahendusi. Esimeses alampeatükis on rohkem administratiivne tegevus, seega on see oluliselt väiksema mahuga, teine alampeatükk sisaldab rakenduse programmeerimist.

3.1 SharePointi sait ja loend

Alguses loodi SharePointi sait. Selleks tuli minna SharePointi halduskeskusesse ning saitide loendis luua uus sait. Valida tuli kahe tüübi vahel: meeskonnatöö või teabeedastussait. Otsus tehti esimese kasuks, kuna see on mõeldud väiksemale kogukonnale. Saidi nimeks valiti „Personal“, sest sinna tulevad ka muud personaliga seotud andmed, mitte ainult käesoleva lõputöö lahendus. Peale saidi loomist loodi tagataustal tegelikult Microsoft 365 rühm, millega on seotud SharePointi sait. Seega saab väga lihtsalt hallata, et kellel on loodud saidile ligipääs.

Järgmisena loodi SharePointi loend, kus hoitakse töötajate Azure AD kontode andmeid. Loendit on võimalik luua kasutades Microsofti poolt tehtud malle, kuid loodavale lahendusele ei ole ükski nendest sobiv, sest vastav mall lihtsalt puudub, seega tehti kõik väljad käsitsi. Loend loodi *Employees* nimega ning järgmiste väljadega: *FirstName*, *LastName*, *JobTitle*, *Country*, *BirthDay*, *Picture*, *OfficePhone*, *DisplayName*, *OfficeLocation*, *HireDate*, *UserId*, *FullName*, *Title*, *createdDateTime*, *lastModifiedDateTime*, *createdBy*, *lastModifiedBy* ja *id*. Viimased 6 välja loob ja haldab SharePoint automaatselt. Eelpool nimetatud väljade nimed on sisemised, kuvatavad väärtused muudeti loetavamaks. *UserId* ja *id* vahe on selles, et esimene on Azure AD konto ja teine on SharePointi loendi kirje identifikaator. Riigi veerule lisati vormindamine JSON(*JavaScript Object Notation*) kujul, mis kuvab riigi nime asemel riigilipu pildi (joonis 10).

```

1  {
2  "$schema": "https://developer.microsoft.com/json-schemas/sp/
   column-formatting.schema.json",
3  "elmType": "div",
4  "children": [
5    {
6      "elmType": "img",
7      "attributes": {
8        "src": "='https://flagpedia.net/data/flags/h20/'+if(@currentField ==
   'Estonia','ee',if(@currentField == 'Norway','no',if(@currentField ==
   'Poland','pl',''))+'.png'",
9        "title": "@currentField"
10     }
11   }
12 ]
13 }

```

Joonis 10. Riigi veeru vormindus JSON kujul.

Lisaks seadistati veel loendi kirjade versioneerimine, mis näitab viimased 50 muudatust iga kirje kohta.

3.2 Tagarakendus

Tagarakenduse loomiseks kasutati Visual Studio 2019, kuhu on paigaldatud Azure'i arenduse tööriistad, mis on Azure Functionsi jaoks vajalikud. Projektis on kasutusel Azure Functions v3 ja .NET Core 3.1. Kuna Azure Functionsi .NET 5 tugi ei olnud sellel hetkel kõige parem, näiteks pidi manuaalselt siluja kinnitama vastava protsessi külge, seetõttu otsustati vanema versiooni kasuks [17].

3.2.1 Projekti tooriku loomine ja Azure'i seadistamine

Projekt loodi Visual Studios kasutades Azure Functionsi HTTP käivituse malli. See tekitas .NET Core 3.1 projekti, kus oli C#'i fail, mis sisaldas HTTP käivituse funktsiooni, "host.json" faili lisati parameeter, mis muutis funktsiooni maksimaalse kestvuse 10 minuti peale, varasema 5 asemel ning "local.settings.json" failis asuvad lokaalsed seaded. Pilve seaded asuvad Azure'is vastavas funktsiooni rakenduses, kus neid saab lihtsalt hallata.

Rakenduse pilves hoidmiseks oli vaja Azure'i tellimust, kuna see oli konkreetse ettevõtte poolt juba antud, siis selleks lisategevusi ei tehtud. Alguses tekitati Azure'i uus ressurssi grupp, sinna sisse loodi uus funktsiooni rakendus ning hoiustamise konto, kuna see on kohustuslik. Plaaniks valiti tarbimise plaan ning asukohaks Põhja-Euroopa. Peale neid

tegevusi tuli vajutada ainult avalikustamise nuppu ning rakendus oli paigutatud pilve. Kogu see protsess tehti läbi Visual Studio.

3.2.2 Microsoft 365 integratsioon

Järgmisena loodi Microsoft 365 integratsioon, mida oli vaja selleks, et sealt andmeid pärida, muuta ja sisestada. Otsustati kasutada Microsoft Graph API't, mis on mõeldud Microsofti pilveteenuste ressurssidele ligipääsuks [18]. Eksisteerivad ka teised API'd, aga siis peaks kasutama neid mitu, näiteks SharePoint ja Azure AD Graph API. Viimasel nendest eemaldatakse tugi täielikult 30.06.2022 [19]. Seega on Microsoft Graph API mõistlik valik.

Alguses siseneti Azure'i portaali ja sealt valiti Azure AD. Järgmisena navigeeriti rakenduste registreerimise lehele, kuhu lisati uus rakendus. Peale nime valiku tuli seal määrata toetatud kontotüübid. Valiti, et ainult kontod selle organisatsiooni kataloogist, kuna teistel seda kasutada pole vaja. Edasi liiguti API õiguste vaatesse, kuhu lisati Graph API'le järgmised õigused:

- GroupMember.Read.All – grupi liikmete lugemiseks,
- Sites.Manage.All – SharePointi loendi kirjade lugemiseks, kirjutamiseks ja muutmiseks,
- User.ReadWrite.All – Azure AD kontode muutmiseks ja lugemiseks.

Eelnevalt tuli valida, kas need on delegeeritud või rakenduse õigused. Valiti viimase kasuks, kuna kasutajad sellesse rakendusse sisse ei logi. Järgmisena genereeriti rakenduse parool.

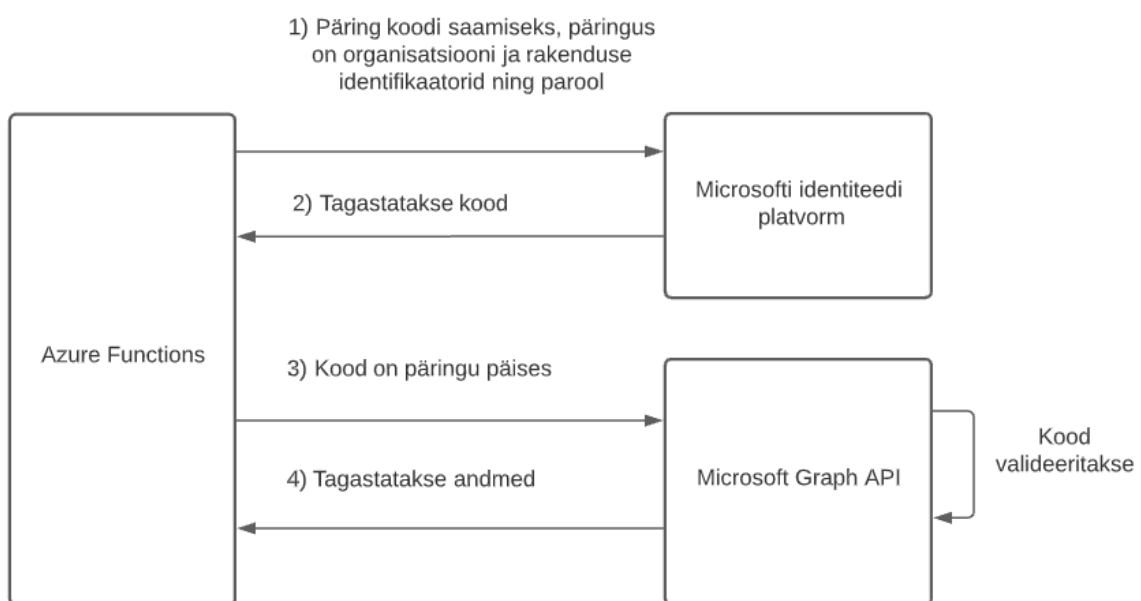
Microsoft on loonud Graph API jaoks SDK(*Software Development Kit*), mis teeb päringute kirjutamise lihtsamaks, loetavamaks ja kiiremaks, selleks paigaldati Microsoft.Graph NuGet pakk. Andmetele ligi pääsemiseks oli vaja koodi Microsoft identiteedi platvormilt, selle mugavamalt saamiseks kasutati Microsoft.Identity.Client NuGet pakki. Koodi lisamiseks tekitati klass, mis implementeerib *IAuthenticationProvider* liidest. Sellel on üks meetodi definitsioon, mille sisendiks on *HttpRequestMessage* tüüpi objekt ning ei tagasta midagi, seal meetodis lisati päringu päisesse vastav kood. Koodi saamiseks oli vaja Azure'is eelnevalt registreeritud

rakenduse ja organisatsiooni identifikaatorit ning selle rakenduse parooli. Kõik need parameetrid võeti Azure'ist ning lisati lokaalsetesse seadetesse ja funktsiooni rakenduse konfiguratsiooni, viimane asus Azure'is. Veel oli vaja skoopi, mis näitab, millised õigused sellel koodil on. Skoobi väärtuseks valiti `https://graph.microsoft.com/.default`, esimene pool sellest näitab, et millisele API'le ligipääsu tahetakse ning `.default` määrab, millised õigused sellel on. See väärtus näitab, et õigused tuleb võtta Azure AD's registreeritud rakendusest, mis eelnevalt seadistati [20]. Kuigi kood ei kesta igavesti ning sellel on aegumise kuupäev, siis loogikat, mis tegeleb koodi uuendamisega ja selle haldamisega, autor kirjutama ei pidanud, kuna koodi saamiseks kasutati eelnevalt mainitud `Microsoft.Identity.Client` NuGet pakki ning see teeb kogu kirjeldatud töö autori eest ära [21]. Järgmisena loodi `GraphServiceClient` tüüpi objekt, mille konstruktorisse lisati koodi saamise klassi objekt. Nüüd on võimalik teha päringuid lihtsamalt ja loetavamalt (joonis 11). Ilma SDK'ta oleks pidanud kõik lõpppunktid sõnena kirjutama, mis oleks nõudnud oluliselt rohkem aega ning vigade tekkimise tõenäosus oleks olnud märgatavalt suurem.

```
await msGraph.Users.Request().GetAsync();
```

Joonis 11. Kasutajate pärimine kasutades Microsoft Graph SDK'd.

Järgmisel joonisel (joonis 12) on näha koodi saamise ning Microsoft Graph API'ga andmete pärimise voog [22].



Joonis 12. Microsoft Graph API'ga andmete pärimine.

3.2.3 Masspäringud

Microsoft Graph API'l on enda süsteem päringute piiramiseks. Kui teha liigselt päringuid, siis mõneks ajaks tagastatakse veateade koodiga 429 ja aeg, millal piirang läbi saab. Võimalik on ka selline olukord, kus lugemine on lubatud, aga kirjutamine mitte. SharePointis ei ole need piirangud kindlalt paika pandud, vaid Microsoft jälgib ressursside kasutust ning vastavalt sellele reguleerib limiite, sest vastasel juhul oleksid piirangud liiga ranged [23], [24].

Piirangute rakendamise tõenäosuse vähendamiseks otsustati kasutada masspäringuid. Nendega on võimalik mitu päringut kokku panna üheks ning tagasi saadetakse üks vastus, seega vähendab see oluliselt päringute arvu, korraga saab saata maksimaalselt 20 päringut. Ühes masspäringus ei pea olema sama lõpppunktiga päringud ning on võimalik öelda, millises järjekorras neid täidetakse [25].

Loodi C#'i klass, kus on 2 välja ja 3 meetodit. Üks väli on Graph API klient ja teine, *batchRequestContent*, hoiab endas päringuid, mida saatma peab. Üks meetod saadab masspäringu, kasutades POST meetodit ning tagastab vastuse. Ülejäänud kahe meetodi puhul kasutatakse ülelaadimist, see tähendab, et meetodite nimed on samad, aga sisendparameetrid või tagastatav tüüp on erinev. Need meetodid lisavad päringu *batchRequestContent* välja ning juhul, kui seal on maksimaalne arv päringuid, siis kasutatakse esimesena kirjeldatud meetodit ja tehakse masspäring ning tühjendatakse vastav väli päringutest. Päringute lisamiseks kasutati kahte meetodit sellepärast, et ühe sisendparameeter oli *HttpRequestMessage* ja teise *BatchRequestStep* tüüpi objekt. Nende vahe seisneb selles, et viimase variandiga on võimalik kaasa anda unikaalne identifikaator, mille järgi saab leida, et milline vastus konkreetse päringu puhul tagastati. Kokkuvõtteks aitab see teha maksimaalselt 20 korda vähem päringuid.

3.2.4 Azure AD kontode sünkroniseerimine SharePointi loendisse

Esimesena loodi funktsioon, mis sünkroniseerib Azure AD kontode andmed SharePointi loendisse. Alguses seadistati käivituse tüübiks HTTP päring, kuna nii oli lihtsam testida, hiljem muudeti see ümber.

Alguses päriti kõik kasutajad Microsoft 365 grupist, kus olid ettevõtte töötajad. Liikmete pärimiseks oli vaja grupi identifikaatorit, mille saamiseks kasutati Microsoft Graph API Explorerit, millega saab Graph API päringuid teha, sest see oli kiireim viis. Saadud identifikaator salvestati nii lokaalselt, kui ka Azure'is, seda tehti ka kõikide järgnevate vajaminevate seadetega. Grupi liikmete pärimisel ei olnud kasutajatel piisavalt andmeid küljes. Seega tuli kasutajad uuesti küsida koos vajalike parameetritega. Alguses prooviti kasutada lõpppunkti, mis tagastaks ühe päringuga kõik kasutajad, mille seast oleks saanud õiged välja filtreerida, sest vajalike kasutajate identifikaatorid olid olemas, need saadi eelnevalt grupi liikmete pärimisel. Seda meetodit ei saanud kasutada, sest sünnipäeva välja selle lõpppunktiga ei olnud võimalik pärida. Seetõttu tuli kõik kasutajad, kes olid vastavas Microsoft 365 grupis, eraldi pärida. Kasutati *select* parameetrit, millega määrati, millised andmed tagastatakse. See aitab pärida täpselt nii palju andmeid, kui vaja läheb. Kasutajate saamiseks kasutati masspäringuid.

Järgmisena päriti kõik kirjed SharePointi loendist. Selleks kasutati samamoodi *select* parameetrit, et tagastatakse ainult nende väljade andmed, mida vaja läheb ning *top* parameetrit, millega seadistati maksimaalne kirjete arv ühel lehel. Kuna tagastati objekt, kus olid loendi andmed ning sõnastiku väli, kus olid kirjed, siis loodi uus staatiline klass, mis konverteeris tagastatud objekti sõnastiku *Employee* tüüpi objektiks. *Employee* klass loodi autori poolt, seal olid olemas kõik vajalikud töötaja väljad. Ainukene väärtus, mida tehtud päringuga kätte ei saadud oli profiilipilt, saadi ainult selle faili nimi. Selleks tehti eraldi toimetused. Alguses sisestati manuaalselt loendi kirjele pilt, et näha, kuhu see salvestatakse. See salvestati töötajate loendi identifikaatori nimelisse kausta, mis asus loendite kaustas, mis omakorda asus saidi varade kaustas. Nagu ka grupi identifikaatori leidmisega, tehti sama ka vastavate kaustadega ning need salvestati samamoodi. Kuna faili nimi oli olemas, siis tehti päring, et saada pildi identifikaator. Kõik pildid olid samas kaustas, mis tähendas seda, et ei saa tekkida sellist probleemi, et on 2 samanimelist pilti. Kasutades pildi identifikaatorit, tehti päring vastavasse lõpppunkti ja saadi pildi *Stream* tüüpi objekt, mis lisati *Employee* tüüpi objektile. Järgmisena võrreldi grupi liikmeid ja SharePointi loendi kirjeid ning kui loendis oli töötaja, mida grupis ei olnud, siis see kustutati loendist ära. Lahkunud töötajate andmeid ei ole mõtet loendis hoida, sest keegi neid ei uuenda.

Edasi itereeriti üle kõigi grupi liikmete ning igale kontole seati vastav loendi kirje. Kui vastavat kirjet ei leitud, siis tekitati uus kirje. Selleks päriti algul konto profiilipilt, sest

nagu loendi kirjel, ei tulnud ka kasutajal see väärtus kaasa. See käis lihtsalt, oli vaja ainult konto identifikaatorit ja teha päring. Kui profiilipilt oli olemas, siis päriti lisaks pildi metaandmed, sealt oli vaja faili laiendit. Profiilipildi üleslaadimiseks SharePointi tehti PUT päring ning kaasa anti faili asukoht ja nimi koos laiendiga ning pildi sisu *Stream* tüüpi objektina. Peale pildi üleslaadimist tekitati loendi kirje koos vastavate väljade väärtustega.

Kui vastav loendi kirje leiti, siis töötaja Azure AD kasutajat uuendati. Selleks päriti töötaja profiilipilt, nagu eelnevalt kirjeldatud. Järgmisena võrreldi kasutaja ning SharePointi loendis olevaid välju ning kui mingi väli ei olnud sama, siis lisati see sõnastikku. Piltide väärtusi võrreldi Base64 vormingus, selleks kirjutati *Stream* klassile juurde vastav laienduse meetod, mis konverteerib selle Base64 vormingusse. Kui pildid olid erinevad, siis *Stream* tüüpi objekti asukoht seati tagasi algusesse, sest seda ei saa muidu uuesti lugeda, mida oli vaja teha pildi üleslaadimisel. Muudeti ainult erinevad väljade väärtused, kuna ei oleks mõistlik kogu aeg kõiki välju uuendada.

Peale esimest sünkroniseerimist oli näha, et väga suur hulk töötajaid ei uuenda enda kontot üldse. Seetõttu kasutati taimeri käivitust, mis käivitab funktsiooni ühe korra nädalas.

3.2.5 Azure AD kontode uuendamine SharePointi loendis olevate kirjete põhjal

Järgmisena loodi funktsioon, mis uuendas töötajate Azure AD kontosid SharePointi loendis olevate andmete põhjal. Testimiseks kasutati HTTP päringu käivitust, kuid hiljem muudeti see taimeri, mis käivitab funktsiooni korra päevas, peale. Esimesena päriti loendist kõik kirjed, mida on viimase 24 tunni jooksul muudetud, sest funktsioon käib iga selle aja tagant. Iga leitud kirje kohta päritakse vastav Azure AD konto. Loendi ja kontode andmete pärimisel kasutati samu meetodeid, mis eelnevalt kirjeldatud.

Konto uuendamiseks pidi päringule kaasa andma *User* tüüpi objekti. Seega on 2 valikut:

- anda igale väljale vastav väärtus,
- lisada kõik väärtused *AdditionalData* välja. See on sõnastik, mille võtmeks on välja nimi ning väärtuseks selle välja uus väärtus.

Alguses prooviti esimest varianti, kuid sellega ilmnes probleem, et kui uuendatav väärtus oli tühi, siis tegelikult välja ei uuendatud ning ühtegi veateadet ei tagastatud. Seetõttu otsustati teise variandi kasuks. Sõnastikku lisati ainult väljad, mida oli vaja uuendada. Kui oli vaja profiilipilti uuendada, siis algul uuendati see Azure AD kontol, järgmisena päriti uue pildi väärtus ning laaditi üles SharePointi loendisse. Seda tehti sellepärast, et sama pilt oleks mõlemas kohas, kuna konto uuendamisel muudetakse pildi parameetreid automaatselt, siis loendis võib see teistmoodi välja näha.

4 Tulemused ja edasiarendus

Käesolevas peatükis kirjeldatakse rakenduse vastavust algselt seatud nõuetele ning võimalikku edasiarendust. Lõputöö käigus loodi valmislahendus, mille konkreetne tarkvaraarendusega tegelev ettevõtte otsustas kasutusele võtta. Rakendust (Lisa 2 – Rakenduse lähtekood) testiti selle loomise käigus manuaalselt ehk eraldi teste selle jaoks ei kirjutatud.

Töö eesmärgiks oli luua lahendus, kus personalitöötajad saavad hallata töötajate Azure AD kontosid selliselt, et haldajatele ei peaks lisama juurde ühtegi Azure AD õigust. Selleks loodi SharePointi sait, see tagas ligipääsetavuse haldamise, kus asus töötajate andmete loend. Tagarakenduseks tehti Azure Functions, kasutades C# programmeerimiskeelt. See koosneb kahest taimeri käivitusega funktsioonist, mis sünkroniseerivad töötajate kontode andmed loendisse ja vastupidi. Microsofti teenustega suhtlemiseks kasutati Microsoft Graph API't rakenduste õigustes. See tagas selle, et personalitöötajatele ei pea lisama juurde ühtegi Azure AD õigust, sest need antakse rakendusele endale. SharePointi loendis on võimalik hallata ainult nõuetes seatud kontode välju, otsida ja filtreerida kirjeid erinevate parameetrite järgi ning muuta välju redigeerimise vormil. Nõuetes on kirjas, et töötajad peavad nägema üksteise telefoninumbreid. Kuna lahenduses on võimalik muuta ka telefoninumbreid, siis need ilmuvad automaatselt näiteks Outlookis profiili vaadates ning sellega on antud punkt täidetud.

Rakenduses on kasutatud õigusi, mis annavad õigused terve organisatsiooni piires. Näiteks, et töötajate loendist andmeid lugeda, on rakendusel tegelikult õigus lugeda kirjeid ka kõikidest teistest loenditest ja muudest ressurssidest. Selle probleemi lahendamiseks tuli Microsoft hiljuti välja lahendusega, kus õiguseid on võimalik anda saidikogumite põhised, selleks tuleks registreerida uus rakendus, anda vastavad õigused ning siis läbi selle rakenduse anda õigused teistele rakendustele [26]. See lahendus kavatsetakse loodud rakenduses kasutusele võtta. Lisaks, kui ettevõttel peaks tekkima soov hallata ka teisi kontode välju, siis saab need enamasti lihtsa vaevaga juurde lisada, kuna suur osa väljadest on sõne kujul.

Veel on kavas luua SharePointi veebiossa, mis kuvab valitud aastal ja kuul töötajate tähtpäevi, näiteks katseaja lõpud, sünnipäevad ja tööjuubelid. See ei ole küll otseselt

käesolevas töös loodud lahenduse edasiarendus, kuid need on omavahel seotud, sest veebiosa kasutaks andmeid, mis Azure Functions töötajate loendisse sünkroniseerinud on.

5 Kokkuvõte

Käesoleva töö tulemusena loodi lahendus, kus personalitöötajatel on võimalik hallata töötajate Azure AD kontosid selliselt, et haldajale pole vaja lisada juurde Azure AD õiguseid. See oli ka lõputöö eesmärk, mis sai edukalt täidetud, kuna nõuded olid täidetud ning ettevõtte võttis lahenduse kasutusele. Töö tulemusena ei pea personalitöötajad enam töötajate andmeid vana siseveebi kaudu haldama.

Lahendus koosneb SharePointi saidist, kus asub töötajate loend ning tagarakenduseks loodi Azure Functions, kasutades C# programmeerimiskeelt. Loendi kaudu saavad personalitöötajad muuta seal olevaid andmeid ning tagarakendus sünkroniseerib uuendused iga teatud aja tagant Azure AD kontodele.

Rakenduses on kasutatud organisatsiooniüleseid õiguseid, kuid tulevikus on see kavas üle viia saidikogumi põhiseks. Veel on plaanis luua SharePointi veebiossa, mis kuvab valitud kuul olevaid töötajate tähtpäevi.

Kasutatud kirjandus

- [1] How user profile synchronization works in Office 365 services.
<https://blog.atwork.at/post/How-user-profile-sync-works-in-Office-365-services#:~:text=Office%20365%20users%20can%20update,user%20can%20update%20specific%20properties> (27.01.2021)
- [2] Microsoft 365 Developer Program subscription expiration and renewal.
<https://docs.microsoft.com/en-us/office/developer-program/subscription-expiration-and-renewal> (27.01.2021)
- [3] Exchange Online. <https://docs.microsoft.com/en-us/Exchange/exchange-online> (27.01.2021)
- [4] user resource type. <https://docs.microsoft.com/en-us/graph/api/resources/user?view=graph-rest-1.0> (27.01.2021)
- [5] Azure AD'ga integreeritud keskkonna loomine töötajate kontaktide haldamiseks SharePointis. <https://digikogu.taltech.ee/et/Item/08ff3cec-0eed-4342-9041-4bf68a5e68ba> (27.01.2021)
- [6] Licensing Updates to Microsoft Flow and PowerApps in Office 365.
<https://powerusers.microsoft.com/t5/Power-Automate-Community-Blog/Licensing-Updates-to-Microsoft-Flow-and-PowerApps-in-Office-365/ba-p/191196#> (27.01.2021)
- [7] Update subscription. <https://docs.microsoft.com/en-us/graph/api/subscription-update?view=graph-rest-1.0&tabs=http> (11.02.2021)
- [8] Why use serverless computing?
<https://www.cloudflare.com/learning/serverless/why-use-serverless/> (11.02.2021)
- [9] Introduction to Azure Functions <https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview> (05.03.2021)
- [10] Azure Functions Pricing. <https://azure.microsoft.com/en-us/pricing/details/functions/> (05.03.2021)
- [11] Azure Functions hosting options <https://docs.microsoft.com/en-us/azure/azure-functions/functions-scale> (05.03.2021)
- [12] What is AWS Lambda?
<https://docs.aws.amazon.com/lambda/latest/dg/welcome.html> (05.03.2021)
- [13] Invoke AWS Lambda Functions Over HTTPS <https://aws.amazon.com/about-aws/whats-new/2015/07/invoke-aws-lambda-functions-over-https/> (10.03.2021)

- [14] Tutorial: Schedule AWS Lambda Functions Using CloudWatch Events
<https://docs.aws.amazon.com/AmazonCloudWatch/latest/events/RunLambdaSchedule.html> (11.03.2021)
- [15] AWS Lambda Pricing <https://aws.amazon.com/lambda/pricing/> (11.03.2021)
- [16] Lambda quotas <https://docs.aws.amazon.com/lambda/latest/dg/gettingstarted-limits.html> (11.03.2021)
- [17] Develop and publish .NET 5 functions using Azure Functions
<https://docs.microsoft.com/en-us/azure/azure-functions/dotnet-isolated-process-developer-howtos?tabs=browser&pivots=development-environment-vs>
(20.03.2021)
- [18] Overview of Microsoft Graph <https://docs.microsoft.com/en-us/graph/overview>
(21.03.2021)
- [19] App migration planning checklist <https://docs.microsoft.com/en-us/graph/migrate-azure-ad-graph-planning-checklist> (21.03.2021)
- [20] Get access without user <https://docs.microsoft.com/en-us/graph/auth-v2-service>
(24.03.2021)
- [21] Daemon app that calls web APIs – acquire a token
<https://docs.microsoft.com/en-us/azure/active-directory/develop/scenario-daemon-acquire-token?tabs=dotnet> (27.03.2021)
- [22] Microsoft identity platform and the OAuth 2.0 client credentials flow
<https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-client-creds-grant-flow> (28.03.2021)
- [23] Microsoft Graph throttling guidance <https://docs.microsoft.com/en-us/graph/throttling#throttling-and-batching> (29.03.2021)
- [24] Avoid getting throttled or blocked in SharePoint Online
<https://docs.microsoft.com/en-us/sharepoint/dev/general-development/how-to-avoid-getting-throttled-or-blocked-in-sharepoint-online> (29.03.2021)
- [25] Combine multiple requests in one HTTP call using JSON batching
<https://docs.microsoft.com/en-us/graph/json-batching> (29.03.2021)
- [26] Controlling app access on a specific SharePoint site collections is now available in Microsoft Graph <https://developer.microsoft.com/en-us/graph/blogs/controlling-app-access-on-specific-sharepoint-site-collections/> (10.04.2021)

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Tair Kokk ----

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Töötajate Azure AD kontode haldus SharePointis“, mille juhendaja on Aleksei Talisainen
 - 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.

14.05.2021

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

Lisa 2 - Rakenduse lähtekood

<https://github.com/tako-0/EmployeesSync>