TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Taavi Ansper 185207IADB

Ettevõtte personaliinfo portaali arendus

Bakalaureusetöö

Juhendaja: Toomas Lepik
MSc

Kaasjuhendaja: Tarmo Oja
MSc

Tallinn 2023

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: Taavi Ansper

06.01.2023

Annotatsioon

Bakalaureusetöö eesmärk on täiustada Cybernetica AS sisemises kasutuses olevat personaliinfo portaali, mis toetab ettevõtte äriprotsesse parema informatsiooni liikumise ja läbipaistvusega. Portaal võimaldab ettevõtte töötajatel ühest kohast leida infot töötajate kontaktandmete, grupikuuluvuste, pääsuõiguste ja puhkuste kohta. Töö käigus koguti tagasisidet ning ideid eelmiste versioonide kasutajatelt. Kogutud nõuete alusel valmis personaliinfo portaali uus versioon. Suurima muudatusena võeti kasutusele uus liidestus Personal365 teenusega. Portaali arendamise ja evituse hõlbustamiseks võeti kasutusele GitLab CI/CD lahendus, mis ehitab Dockeri kettapildid automaatselt. Dockeri kettapiltide evitamiseks on kasutusel konfiguratsioonihaldustarkvara Ansible.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 7 peatükki ja 15 joonist.

Abstract

Development of a corporate personnel information portal

The purpose of the thesis is to enhance the personnel information portal that is in use for the internal use of Cybernetica AS employees. The portal supports the corporations business processes thanks to better information flow and transparency. The portal allows employees of the corporation to find information about employees contact information, group memberships, access rights and vacations. During the development phase feedback and ideas were gathered from the users of the previous versions of the portal. From the requirements that were collected, a new version of the personnel information portal was developed. The biggest change in the new version was the integration of Personal365 services. For the development and deployment of the portal, Gitlab CI/CD was taken into use, which is used to build new Docker images of the application. For the deployment of the Docker images, Ansible configuration management software is used.

The thesis is written in Estonian and is 29 pages long, including 7 chapters and 15 figures.

Lühendite ja mõistete sõnastik

ACL	*Access Control List* - pääsuloend
API	*Application Programming Interface* - rakendusliides
CI/CD	*Continuous Integration / Continuous Delivery* - Pidev integratsioon / pidev tarne
ERP	*Enterprise Resource Planning* - ettevõtte ressursside planeerimine
HTTPS	*Hypertext Transfer Protocol Secure* - turvaline hüperteksti edastuse protokoll
JPEG	*Joint Photographic Experts Group* - fotospetsialistide ühisrühm
JSON	*JavaScript Object Notation* - JavaScripti objektide notatsioon
LDAP	*Lightweight Directory Access Protocol* - Kataloogipöörduse kergprotokoll
SCP	*Secure Copy* - turvaline kopeerimine
SSH	*Secure Shell* - turvaline kest
TLS	*Transport Layer Security* - transpordikihi turve
URL	*Uniform Resource Locator* - ühtne ressursilokaator
VM	*Virtual Machine* - virtuaalmasin
VPN	*Virtual Private Network* - virtuaalne privaatvõrk
WAF	*Web Application Firewall* - veebitulemüür

Sisukord

Jooniste loetelu

1. Sissejuhatus

Personaliinfo portaal on veebileht, kus kasutajad saavad näha enda ja teiste kasutajate, ehk kolleegide, ettevõttesiseselt avalike andmeid. Näiteks kui Mari tahab vaadata Peetri telefoninumbrit, siis ta läheb personaliinfo portaali ja otsib sealt Peetri profiili. Teisalt kui Peeter tahab vaadata, mis ressursidele saab ligi tema osakond, siis ta läheb personaliinfo portaali, avab LDAP gruppide vaate, otsib sealt oma osakonda nimepidi ja leiab ressursid, millele ta osakond ligi saab. Need on kaks olulist personaliinfo portaali funktsiooni.

Personaliinfo portaalist on kasutusel olnud mitu versiooni. Esimene versioon täitis esimeses näites toodud funktsionaalsust hästi, aga puudu oli funktsionaalsus, mis oli välja toodud teises näites. Personaliinfo portaal teine versioon on selle lõputöö autori kirjutatud. Teise versiooni eesmärk oli parendada esimest versiooni ja tuua juurde funktsionaalsust, mis oli puudu algses versioonis.

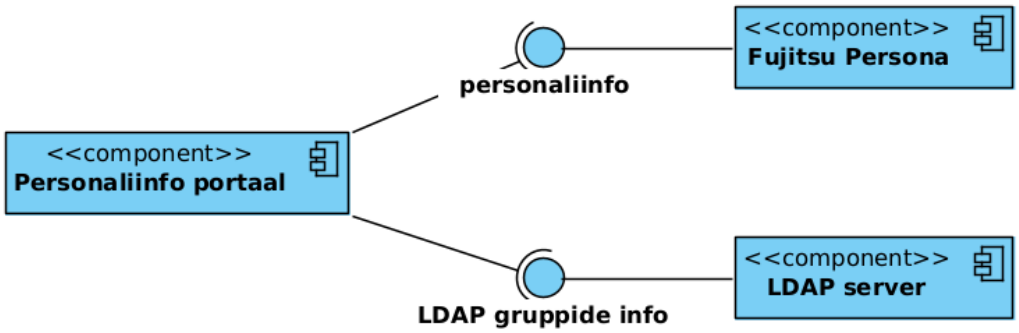
Lõputöö käsitleb personaliinfo portaali uue, kolmanda, versiooni arendust. Personaliinfo portaal saab kasutajate andmed erinevate teenuste käest: personalihalduse teenusest, LDAP-i serverist, töövoohalduse teenusest, DokuWikist ja failiserverist. Peamised kaks teenust on personalihalduse teenus, mis sisaldab töötajate üldandmeid, ning LDAP-i server, mis sisaldab ettevõtte töötajate kasutajakontode informatsiooni. Kuna Cybernetica asendas Fujitsu Persona personalihaldustarkvara Microsoft Business Central ERP lahendusega, tekkis vajadus arendada välja uus versioon personaliinfo portaalist. Uut versiooni arendades tekkis ka võimalus ära parandada vead, mis tulid välja kasutajate tagasisidest, kes kasutasid teist versiooni portaalist. Portaali jätkusuutlikuma arenduse jaoks on vaja parendada rakenduse evitamise töövoogu, mis hõlmab automaatsemat rakenduse ehitusprotsessi ja tehiseid mida saab mugavalt evitada.

Töö esimeses pooles vaadeldakse personaliinfo portaali eelmiseid versioone ja nende funktsionaalsust. Järgmisena kogutakse nõudeid portaalile ja tehakse ohtude kaardistus kasutades STRIDE-LM[1] mudelit ning on juttu tehnoloogia valikutest. Järgmised peatükid räägivad nõuetele vastava kasutajaliidese tegemisest ning rakenduse arhitektuurist ja arenduse käigus ette tulnud muudatustest. Viimane peatükk räägib rakenduse ehitamisest ja selle evitamisest.

2. Personaliinfo portaali eelnevad versioonid ja nende puudused

Personaliinfo portaalist on kasutusel olnud kaks versiooni. Esimene versioon, mis oli arendatud enne lõputöö autori liitumist ettevõttega ja teine versioon, mis on lõputöö autori arendatud.

2.1 Personaliinfo portaali esimene versioon



Joonis 1. *Personaliinfo portaali esimese versiooni struktuur.*

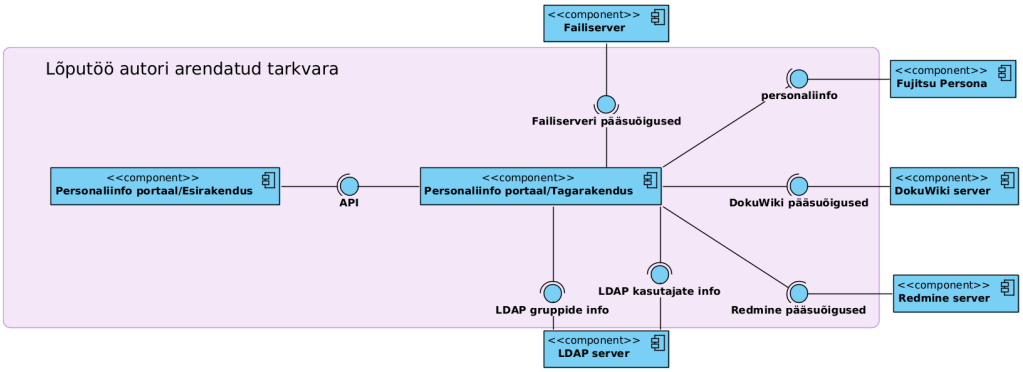
Rakendus arendati enne lõputöö autori ühinemist ettevõttega, kasutusel oli see olnud mitmeid aastaid. See rakendus oli kirjutatud PHP-s. Lähtekood asus serveris, kus seda jooksutati ja lähtekoodi haldamisel ei kasutatud versioonihaldust. Rakendust ei saanud lihtsalt evitada teise serverisse, kuna ei olnud kasutusjuhendit ega dokumentatsiooni. Rakendus kuvas inimeste kohta kõike vajalikku infot, mida Fujitsu Persona API kaudu sai pärida ja see oli rakenduse peamine funktsionaalsus. Personaliinfo portaali esimese versiooni struktuur on kujutatud joonisel [1].

Kuna API otspunkt oli aeglane ja andmete pärimine võttis aega, siis rakenduse kuvatavad andmed salvestati iga päev Linuxi utiliiti cron kasutades failidena failipuusse, kust nende pärimine on kiirem, kui Fujitsu pakutud teenusest.

Rakenduse teine funktsionaalsus, anda infot kasutaja ressurside õiguste kohta, oli puudulik. Selles vaates sai näha kõiki LDAP-i gruppe ja seal olevaid kasutajaid, aga rohkem see ei võimaldanud. Rakenduses oli ka funktsionaalsus üles laadida kasutajate fotosid, mis talletati serveri failipuusse. Need failid seoti kasutades isikukoodi failinimes Personast tulnud andmetega.

2.2 Personaliinfo portaali teine versioon

Personaliinfo portaali teine versioon parandas esimese versiooni suurimad puudused ning lahendas ettevõtte kasvust tekkinud uued vajadused. Eesrakendus (*frontend*) on kirjutatud Typescript keeles, kasutades React raamistikku. Kujunduseks kasutatakse Material UI teeki. Tagarakendus on kirjutatud Pythoni keeles, kasutades Flask raamistikku. Liidestused Redmine-iga on tehtud kasutades Pythoni teeki python-redmine. See personaliinfo portaal kasutab sama API otspunkti töötajate andmete pärimiseks, mis esmane versioon rakendusest ja salvestab samamoodi andmed serveri failipuusse. Personaliinfo portaali teise versiooni struktuur on kujutatud joonisel [2].



Joonis 2. *Personaliinfo portaali teise versiooni struktuur.*

Esimeses versioonis oli organisatsiooni struktuur koodi sisse kirjutatud. Uus versioon võttis osakondade nimetused Persona API kaudu päritud andmetest. Selle tõttu sai mugavamalt muuta osakonna nimetust Persona süsteemis ja ei pidanud iga kord rakenduse lähtekoodi muutma.

Teine versioon rakendusest kuvas kasutaja kohta samu andmeid, mida ka esimene versioon, kuna API kaudu päritud andmete koosseis ei muutunud. Uus rakendus lisas juurde kaks olulist funktsionaalsust:

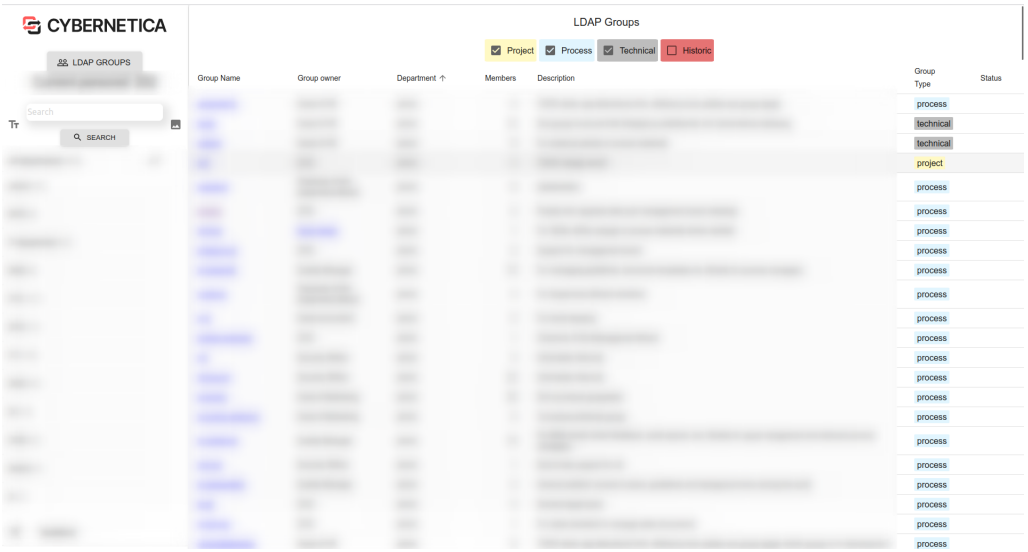
- Kasutajaandmete sidumise LDAP serverist tulnud andmetega ja selle kuvamine kasutajavaates. Kuna LDAP server hoiab endas iga kasutaja ees-ja perekonnanime, siis sai siduda Personast tulnud andmed nende kahe välja abil LDAP serverist tulnud andmetega. Tänu sellele on teises versioonis võimalik näha, millistes LDAP gruppides kasutaja on. Esimene versioon portaalist ei sidunud ära kasutajavaates kasutaja ees- ja perekonnanime LDAP-ist tulnud andmetega.
- Puhkuseinfo kuvamine ja iCal vormingus eksport.

LDAP gruppide vaade on rakenduse teises versioonis tehtud mugavamaks ja sisaldab rohkem infot pääsuõiguste kohta. Projekti käigus korrastati ja dokumenteeriti kõik

grupid. Iga grupi kohta on määratud:

1. Grupi tüüp ("*process*", "*project*", "*technical*")
2. Kas grupp on kasutusel või arhiveeritud (" või "*historic*").
3. Üksus millele grupp kuulub (osakonna nimi või "*historic*").
4. Omanik (kas kasutajatunnus või rollinimi)
5. Kirjeldus

Tänu nendele väljadele saab gruppide kohta palju rohkem informatsiooni, milleks nad kasutusel on. Gruppide vaade rakenduse teises versioonis on kuvatud joonisel [3]. Grupi vaates on näha ka sellele grupile kättesaadavad ressursid eri süsteemides.



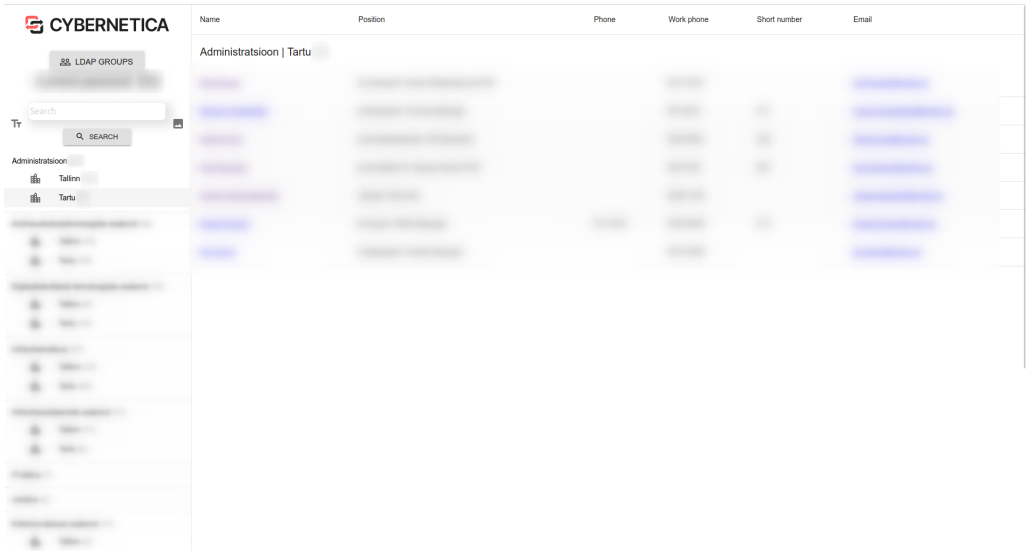
Joonis 3. *LDAP gruppide vaade rakenduse teises versioonis.*

Oluline funktsionaalsus, mis lisati rakenduse teise versiooni on otsinguväli. Seda saab kasutada nii osakondade vaates, kus kuvatakse kasutajaid, kui ka LDAP gruppide vaates.

Lisaks otsingule lisati juurde ka filtreerimine:

1. Osakonna tasandil saab valida osakonna töötajad.
2. Osakkona tasandil saab valida osakonna töötajad kindlas linnas.
3. LDAP gruppide tasandil saab valida mingi osakonnaga seotud LDAP grupid.
4. LDAP gruppide tasandil saab valida grupid vastavalt tüübile.

Filtreerimine on kuvatud joonisel [4].



Joonis 4. *Osakondade vaade filtreeritult.*

3. Nõuded portaalile

Nõuded jagunevad funktsionaalseteks ja mittefunktsionaalseteks. See jaotis kirjeldab personaliinfo portaalile esitatavaid nõudeid.

Nõuete allikad olid:

1. Kasutajate soovid, leitud vead - kasutajad saavad kurta enda muresid ja teha ettepanekuid IT osakonnale, kasutades sisemist töökäskude halduse süsteemi.
2. Arutelud personalitöötajaga.
3. Arutelud kvaliteedijuhiga.
4. Cybernetica integreeritud juhtimissüsteemi juhendid.
5. Cybernetica IT osakonna nõuded ja tavad.
6. Uue ERP süsteemi võimalused, arutelud ERP süsteemi tarnijaga.
7. Töö autori, kui süsteemi haldaja ja kasutaja enda ideed.

3.1 Funktsionaalsed nõuded

Paljud funktsionaalsed nõuded tulid eelmistest personaliinfo portaali versioonidest. Eelmiste süsteemide funktsionaalsus peab jääma alles.

Personaliinfo portaali peamised kasutajad on personaliosakonna töötajad, kes kasutavad seda, et inimeste andmeid vaadata ja see on oluline osa nende tööprotsessist. Nende töötajate suulisest tagasisidest rakendusele saadi sisendit funktsionaalsetele nõuetele.

Oluline sisend portaali nõuetele tuli kasutajatelt. Kasutajad said raporteerida töökäskude süsteemi portaaliga seotud vigu ja soove.

Lõpuks vaatas antud töö autor ise, mis andmed saadaval on ja mis funktsionaalsust saaks veel pakkuda kasutajatele.

3.1.1 Nõuete nimekiri

1. Kasutaja peab saama navigeerida portaali erinevate vaadete vahel.
2. Kasutaja peab saama vaadata ettevõtte töötajate infot tabelina.

3. Kasutaja peab saama töötajaid nime järgi otsida.
4. Kasutaja peab saama filtreerida ettevõtte töötajaid osakondade ja linnade kaupa.
5. Kasutaja peab saama vaadata ettevõtte töötajaid vaates, kus on järjestatud töötajate fotod ja nende all nimed.
6. Kasutaja peab saama vahetada tabelivaate ja pildivaate vahel.
7. Kasutaja peab saama näha nimekirja, kus on kõigi töötajate praegused ja tulevased puhkused.
8. Kasutaja peab saama näha nimekirja, kus on kõikide töötajate sünnipäevad aasta lõikes.
9. Kasutaja peab saama vaadata kasutaja profiili, kus on kirjas kõik selle kasutajaga seotud andmed, sealhulgas ka grupikuuluvused.
10. Kasutaja peab saama vaadata tabelit, kus on loetletud kõik LDAP grupid.
11. Kasutaja peab saama filtreerida LDAP gruppide vahel vastavalt osakonnale, millele need grupid kuuluvad.
12. Kasutaja peab saama filtreerida LDAP gruppide vahel vastavalt grupi tüübile.
13. Kasutaja peab saama filtreerida LDAP-i gruppe vastavalt tabeli päisele tõusvas või alanevas järjekorras.
14. Kasutaja peab saama vaadata LDAP grupi kohta kõiki andmeid, mis on sellega seotud. Sealhulgas juurdepääsu õiguseid teistes süsteemides.
15. Kasutaja peab saama vahetada oma parooli ERP süsteemis.
16. Rakendus peab automaatselt sünkronisseerima LDAP-is kirjeldatud aktiivsed kasutajad ERP süsteemiga.

3.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsed nõuded tulid ennekõike Cybernetica infosüsteemi ülesehitamise ja haldamise tavadest ja vajadustest. Samuti integreeritud juhtimissüsteemi nõuetest. Abimaterjalina on kasutatud Altexsofti blogipostitust[2], kus tuuakse välja erinevad mittefunktsionaalsed nõuded.

Jõudlus Kasutajate hulk, kes rakendust kasutavad on väike: 200 kuni 400. Selletõttu ei pea rakendust tehes arvestama suure kasutajate hulgaga. Samas rakendus peaks kasutaja jaoks laadima kiirelt, rakenduse teise versiooni kohta tuli kasutajatelt tagasisidet, et VPN-i kasutades oli portaali laadimisaeg pikk.

Skaleeritavus Kuna kasutajate hulk on väike, siis rakenduse skaleeritavus ei ole oluline.

Porditavus Cyberneticas on kasutusel serverid ja virtuaalmasinad, mis kasutavad Linuxi distributsioone, valdavas enamuses Debiani, kuna see on tasuta ja stabiilne. Seetõttu peaks rakendus jooksma Linux Debiani peal. Haldamise hõlbustamiseks peab kasutama Dockerit.

Kasutatavus Rakendus peab olema kasutajatele lihtne ja mugav kasutada.

Lokalisatsioon Kuna Cybernetica AS-is on kasutusel nii eesti keel kui ka inglise keel peab rakendus toetama ühte nendest keeltest. Andmed, mida portaalis kuvatakse peavad tõlke olemasolul kuvama mõlemat tõlget.

Töökindlus Rakendus ei ole kriitiline teenus firma tööprotsessis, seetõttu teenuse töökindlus ei ole kõrge prioriteediga.

Kättesaadavus Rakendus ei ole kriitiline teenus firma tööprotsessis, seetõttu teenuse kättesaadavus ei ole kõrge prioriteediga.

Hooldatavus Rakendus peab olema kergesti hooldatav, et vajadusel saaksid ettevõtte teised arendajad muudatusi teha ja neid rakendada. Selletõttu peaksid kasutatavad keeled ja raamistikud olema ettevõtte sees laialdaselt kasutatavad. Lisaks sellele peaks olema ka ettevõtte sees nende keelte kompetents olemas, st. neid on kasutatud projektides ettevõtte sees. Muudatusi peab olema lihtne paigaldada. Rakenduse hõlpsaks uuendamiseks peab pakendamiseks kasutama Dockerit. Rakendus peab ehitama Dockeri kettapildi kasutades CI/CD-d, selleks on olemas Cyberneticas Gitlab. Rakenduse uue versiooni evitamiseks peab kasutama Ansible-t.

Turvalisus Alexasofti nõuete analüüsi üks osa on Turvalisus. Turvalisus on antud rakenduse kõige olulisem osa ja seetõttu on sellest eraldi alamjaotis.

3.3 Turvalisus

Nagu sissejuhatuses on mainitud, kuvab rakendus ettevõtte sees olevaid avalikke andmeid. St. kõikidel töötajatel on õigus nendele andmetele ligi pääseda. Kuna rakendust saab kasutada ainult ettevõtte sisevõrgust, siis enamuse rakenduse funktsionaalsuse kasutamiseks ei ole vaja autentimist. Kuna eelmistes versioonides ei olnud autentimist ja ettevõtte sisevõrk on turvatud tsoon, siis ei ole ka uue rakenduse nõuetes autentimine vajalik nõue.

Rakenduse kõige turvakriitilisem osa on liidestus ERP süsteemiga. Kasutajad ei tohi saada otsejuurdepääsu ERP teenustele. Seetõttu on oluline roll tagarakendusel, mis saab ligi nendele teenustele, kust andmed tulevad. Tagarakendus filteerib neid andmeid ja ebavajalik info eemaldatakse. Kõige tundlikumad andmed asuvad ERP süsteemis, kus asuvad kõikide töötajate isikuandmed, mis on ettevõtte personaliosakonnal vaja töötaja kohta: haridus, pangarekvisiidid, töölepingud jms. Seetõttu on väga oluline, et päringud tehakse läbi tagarakenduse, mis filteerib neid andmeid. ERP rakenduse API dokumentatsioon[3] annab ülevaate kõikidest otspunktidest, kust andmeid pärida. Sealt on olulised otspunktid 'Töötajate loend' [3, Ptk. 5.25, lk 40-41], 'Puhkuste ajakava žurnaal'[3, Ptk. 5.27, lk. 46-47] ja 'Koolituse loend' ning 'Koolitustel osalejad'[3, Ptk. 5.2, 5.3, lk. 7-9].

Kuna tagarakendus küsib erinevatelt teenustelt tundlikke andmeid, on kõik ühendused nende teenustega tehtud üle turvalise ühenduse, isegi kui kõik teenused on sisevõrgus. Kõik päringud API otspunktide pihta kasutavad HTTPS protokolli. Ülejäänud päringud, kus ei ole võimalik kasutada HTTPS protokolli töötavad üle SSH protokolli.

Rakenduse osa, mis lubab kasutajatel muuta oma ERP kasutaja parooli peab olema autenditud, kuna muidu saab ükskõik kes ettevõtte sees muuta kellegi teise parooli. Autentimine käib läbi kasutaja LDAP konto, kuna see on peamine autentimismeetod ettevõtte sees.

Kõik salajased võtmed mida rakendus kasutab, peavad olema turvaliselt hallatud, kasutades Ansible-vault tarkvara.

Rakenduse ohtude kaardistamiseks on kasutatud STRIDE-LM mudelit[1]. Rakenduse ohtude kaardistamise protsess on väga kasulik rakenduse kaitse kindlaks määramiseks. Ohu kaardistamise eesmärk on hinnata süsteemi potentsiaalse ründaja vaatenurgast ja seejärel valida nende rünnakute riski vähendamiseks sobivad lahendused.

STRIDE-LM-i mudeli 7 komponenti on:

1. *Spoofing* ehk teesklus - kasutaja esineb süsteemile teise kasutajana. Rünne on suunatud autentimisfunktsiooni vastu.
2. *Tampering* ehk manipuleerimine - ründaja rikub andmete terviklust eesmärgiga panna andmete kasutaja enda jaoks soodsalt käituma.
3. *Repudation* ehk salgamine - kasutaja eitab mingi tegevuse tegemist, eesmärgiga vabaneda vastutusest.
4. *Information Disclosure* ehk andmeleke - ründaja saab oma valdusesse süsteemis olevad konfidentsiaalsed andmed.
5. *Denial of Service* ehk teenusetõkestus - ründaja tekitab olukorra, kus teised kasutajad ei saa rakendust kasutada.
6. *Elevation of Privilege* ehk õiguste vallutus - ründaja saavutab olukorra, kus tal on süstemis rohkem õiguseid kui ette nähtud.
7. *Lateral Movement* ehk külgliikumine - ründaja kasutab süsteemi ära teiste süsteemide ründamiseks, millede otse ründamiseks tal puuduvad võimalused.

Personaliinfo portaali rakenduse saab jagada kaheks osaks. Avalik osa ja autentimisega osa. Autentimisega osa on ERP süsteemi paroolivahetusleht, mis kasutab LDAP-i serveri vastu autentimist, et vahetada kasutaja parooli ERP süsteemis.

3.3.1 Personaliinfo portaali avaliku osa ohtude kaardistus

Teesklus Juurdepääs rakendusele on piiratud võrgu tasemel. Sisevõrgus rakendus ei autendi kasutajaid, seega teesklusründed pole võimalikud. Ründaja matkib personaliinfo portaali, et anda kasutajale valeinfot. Kaitseks selle ründe vastu on ametliku sertifitseerimiskeskuse poolt väljastatud TLS sertifikaadid, mida ründajal on raske saada.

Manipuleerimine Personaliinfo portaali andmeid manipuleerides on raske asjalikku rünnet korda saata. Kõige tõsisem rünne oleks see, et ründaja peidab vaataja eest LDAP-i gruppide pääsuõiguseid ja vaataja paneb konfidentsiaalse faili kohta, kus ta arvab selle olevat kättesaadava vaid usaldatud kasutajatele, kuid ründajal on ka sinna ligipääs. Kaitsemehhanism on manipuleerimise puhul samagi, mis teeskluse puhul - sertifitseerimiskeskuse poolt väljastatud TLS sertifikaat.

Salgamine Personaliinfo portaali avaliku osa puhul kasutajal puudub interaktsioon, kus on vaja tõestada et midagi tehti. Seetõttu salgamine ei ole personaliinfo portaali puhul oht.

Andmeleke Ettevõte on otsustanud, et personaliinfo portaali kuvatavad andmed on ettevõtte sisene info, millele kõik töötajad peavad ligipääsu saama ja rakendus on evitatud sisevõrku, kuhu pääsevad ainult autenditud kasutajad. Juhul, kui andmed lekivad sisevõrgust välja, siis ohtlik aspekt on pääsuõiguste lekkimine, mille tõttu võib näha kellel on rohkelt õigusi eri süsteemidesse ja läbi selle planeerida rünnakut ja kavandada suhtlusrünnet.

Teenusetõkestus Rakendus on evitatud sisevõrku ja see on loetud piisavaks kaitsemeetmeks.

Õiguste vallutus Kuna rakenduses ei ole eri õiguseid, ehk kõik kasutajad saavad täpselt samu tegevusi teha ja näevad samu andmeid, siis pole ka see rünne võimalik.

Külgliikumine Ennekõike saaks ründaja personaliportaali juurdepääse ära kasutades rünnata ERP süsteemi. Kui ründaja saab sisse virtuaalmasinasse, kus rakendus jookseb, siis saab ligi API võtmele, millega saab ligi ERP süsteemile. Et rakendus saaks ligi võimalikult vähestele muudele teenustele jooksutatakse rakendust Dockeri konteineris ja mitte-juur kasutajana. Lisaks ei ole ERP rakenduse API võti lisatud Dockeri konteinerisse, vaid on eraldi keskkonnamuutuja, mille peab määrama.

3.3.2 Personaliinfo portaali priviligeeritud osa ohtude kaardistus

Teesklus Kuna rakendus kasutab autentimise vastu välist autentimispakkujat ehk LDAP-i, siis teeskluse puhul, saab ründaja ligi antud kasutaja andmetele ERP süsteemis ja näeb ühe kasutaja kohta isikuandmeid, puhkuseinformatsiooni ja ka palgainformatsiooni. Kasutusel on samad kaitsemehhanisimid, mis rakenduse avaliku poole puhul. Kuna tegu on autentimisega, mida peab iga kord kontrollima, siis rakenduses sessioonihaldus ei ole vajalik.

Manipuleerimine Olukord, kus kasutajale määratakse teine parool, kui ta sisestas. Tüütu, aga admin saab selle ära muuta. Kasutusel on samad kaitsemehhanismid, mis rakenduse avaliku poole puhul. Lisaks on kasutusel tekstiväljade kontroll, mis ei luba sisestada potensiaalseid skriptijuppe.

Salgamine Siin saab kasutaja öelda, et ta pole määranud parooli, seda ei pea tõestama. Kasutaja võib uuesti määrata parooli.

Andmeleke Oluline aspekt on, et nii autentimiseks kasutatavat LDAP-i parooli ega seadistatavat ERP parooli ei tohi kuvada ega logida.

Teenusetõkestus Kuna parool määratakse korra ja uuesti määramist on harva, siis teenuse kasutuse maht ei ole suur ja potensiaalne ründevektor ei ohusta kasutamist. Lisaks saab vajadusel administraator määrata käsitsi kasutajatele parooli läbi ERP teenuse enda.

Õiguste vallutus Sama mis rakenduse avaliku osa puhul.

Külgliikumine Sama mis rakenduse avaliku osa puhul.

3.4 Tehnoloogia valik

3.4.1 Programmeerimiskeeled ja raamistikud

Tagarakenduse jaoks on valitud Python[4] ja Flask[5] raamistik. Python on valitud tagarakenduse jaoks mitmel põhjustel. Python on laialdaselt kasutusel olev programeerimiskeel, mida on lihtne õppida. Näiteks, TalTech-i IT erialal on esimene programmeerimise kursus "Programmeerimise algkursus ITI0102", kus kasutatakse Pythonit, et õpetada programeerimise algtõdesid. Lisaks on lõputöö autoril juba eelnev kogemus Pythoniga.

Flask raamistik on valitud kuna see on laialdaselt kasutuselolev ja kergekaaluline raamistik. Lisaks on raamistik kasutusel olnud ettevõtte sees erinevates projektides.

Flask raamistiku õppimiseks on kastuatud Miguel Grinbergi raamatut "Flask Web Development"[6]. See raamat õpetab parimaid tavasid, kuidas arendada veebirakendust kasutades Flaski raamistikku.

Personaliinfo portaali eesrakenduse keeleks on valitud Typescript[7]. Selle eelis Javascripti ees on tüüpimise olemasolu, mis aitab ennetada arendamise käigus vigu. Typescript on Cybernetica AS-is laialdaselt kasutusel programmeerimiskeel.

Eesrakendus kasutab raamistikuna React[8] raamistku. Raamistiku valik on tehtud mitmetel eri põhjustel. Ettevõte erinevates projektides on React laialdaselt kasutusel. Lisaks on lõputöö autoril eelnev kokkupuude antud raamistikuga õppeainest "ICD0006 Javascript".

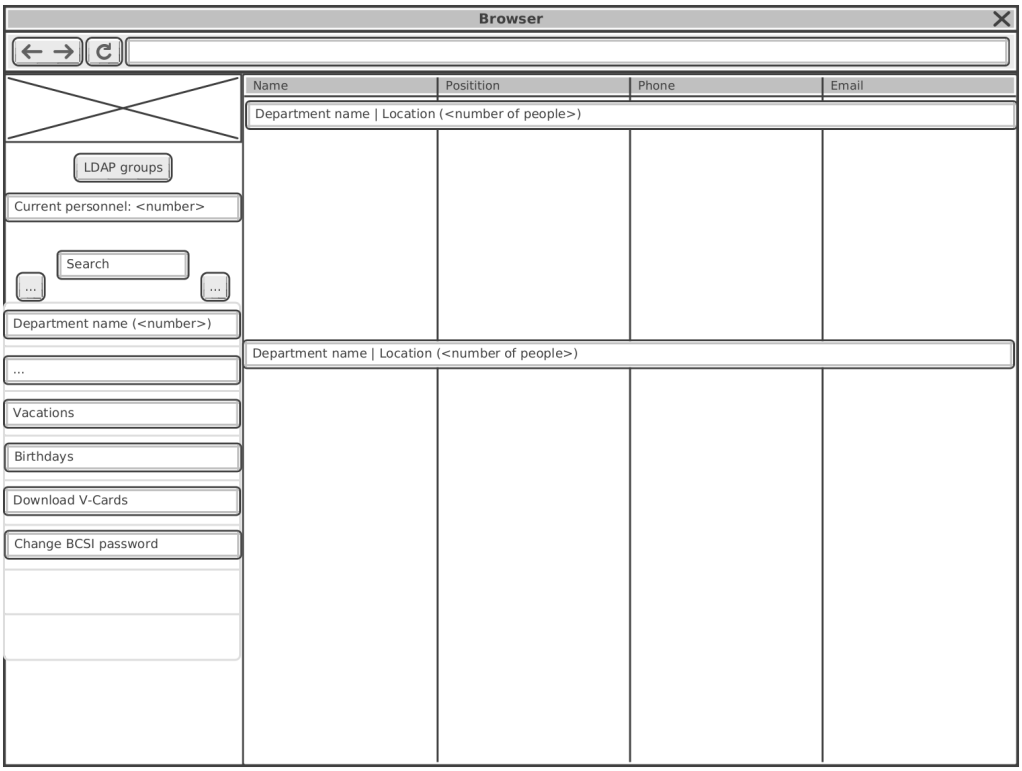
3.4.2 Arendustööriistad

Pythonis ja Javascriptis tarkvara arendamiseks on olulised lisad:

- Pyenv[9] - tööriist, mis lubab alla laadida erinevaid Pythoni versioone ja nende vahel hõlpsalt vahetada. Ei sõltu kindlast Pythoni versioonist, mis on antud keskkonnas kastusel, vaid kasutab ainult Bash-i käsurea skripte. Pyenv lubab ka määrata projektidele kindla Pythoni versioon, mis hõlpsustab kasutajal erinevate projektide vahel liikuda.
- Poetry[10] - tööriist, mis haldab pythoni mooduleid ning asendab Pythoni tööriista pip-i. Poetry asendab tavalise Pythoni arenduse jaoks mõeldud failid: *setup.py, requirements.txt, setup.cfg, MANIFEST.in* ja *Pipfile*. Nende failide asemel on kasutusel ainult üks: *pyproject.toml* fail, mis muudab üldist projekti haldamist mugavamaks.
- NVM[11] - tööriist, mis lubab alla laadida erinevaid NodeJS versioone ja nende vahel hõlpsalt vahetada. NodeJS-ist on palju eri versioone ja kuna projektidel on erinevad versioonid NodeJS-ist kasutusel, siis NVM lubab projektide vahel vahetada versiooni hõlpsalt.

4. Nõuete analüüs

Enamus funktsionaalseid nõudeid on seotud kasutajaliidesega. Nende analüüs ja täpsustamine käis läbi prototüüpimise. Prototüübist sai töö käigus toimiv portaal. Personaliinfo portaali kasutajaliides on jagatud mitmeks osaks. Kasutajaliides koosneb vaadetest, mille sõrestikmudel on kuvatud joonisel [5]:



Joonis 5. *Personaliinfo portaali avalehe sõrestikmudel.*

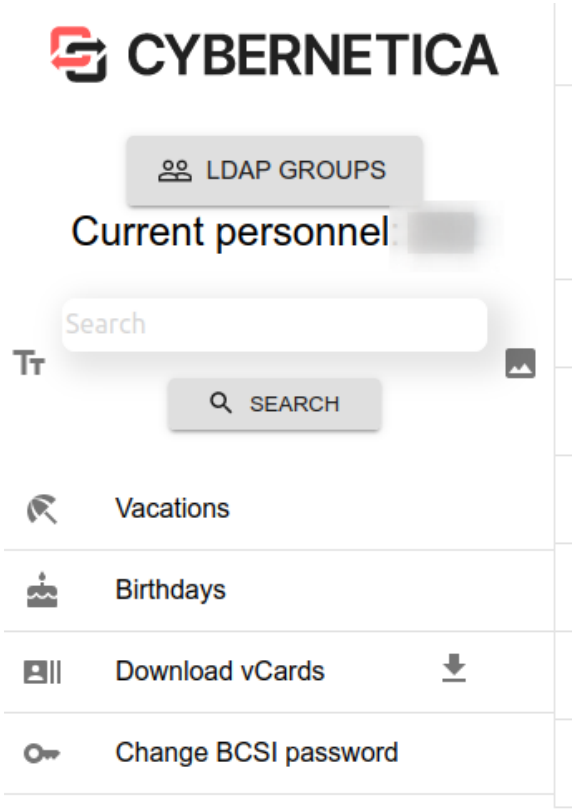
- Kasutajate vaade, mis kuvab tabelit ettevõtte töötajatest.
- Kasutaja vaade, mis kuvab ühe töötaja kohta olevat informatsiooni.
- LDAP gruppide vaade, mis kuvab tabelit kõikidest LDAP-i gruppidest.
- LDAP grupi vaade, mis kuvab ühe LDAP gruppi kohta käivat infot.
- Puhkuste vaade, mis kuvab kõiki puhkusi, mis on planeeritud ja kinnitatud.
- Sünnipäevade vaade, mis kuvab ühe kalendriaasta jooksul olevaid sünnipäevasid.

Vaate vasakus ääres paikneb menüü ehk teise nimega küljeriba.

4.1 Personaliinfo portaali kasutajaliidese komponendid.

Küljeriba

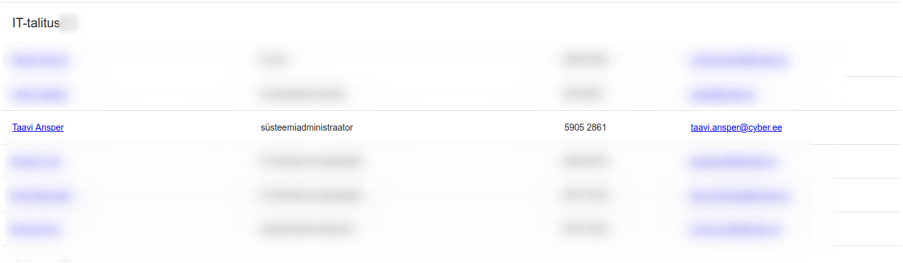
- Rakenduse vasakul küljel on küljeriba, see küljeriba on koguaeg nähtaval, kuna see on menüü, mille kaudu liigeldakse rakenduse eri vaadete vahel. See täidab nõuet nr. 1 peatükist 3.1.1. Küljeriba on kuvatud joonisel [6].
- Küljeriba kuvab:
  1. Ettevõtte logo
  2. Otsinguvälja
  3. Nuppu, mis vahetab osakondade vaadet pildivaate ja tabelivaate vahel.
  4. Nuppu, mis vahetab osakondade vaate ja LDAP gruppide vaate vahel.
  5. Organisatsiooni struktuuri
  6. Nuppu, mis suunab puhkusevaatele.
  7. Nuppu, millega saab alla tõmmata töötajate vCard-id.
  8. Nuppu, mis suunab ERP teenuse paroolivahetus lehele.



Joonis 6. *Küljeriba komponent. Märkus: pildilt on puudu dünaamiliselt nimekirja lisatud organisatsiooni struktuur, mida ei saa kuvada veel, kuna ERP API seda veel ei võimalda.*

Osakondade tabelivaade

- Kasutaja näeb tabelit kõikidest osakondadest ja nendes olevatest kolleegidest. See täidab nõuet nr. 2 peatükist 3.1.1. Osakondade tabelivaade on kuvatud joonisel [7].
- Kasutaja näeb osakondade tabelis enda ja teiste kohta vastavaid kontaktandmeid ja isikuandmeid:
  1. Nimi
  2. Ametikoht
  3. Töötelefon
  4. Meiliaadress
- Vajutades osakonna vaates kasutaja nime peale, suunab rakendus edasi kasutajavaatesse.



Joonis 7. *Osakondade tabelivaade. Lõputöö autori andmed on näitena jäetud udustamata.*

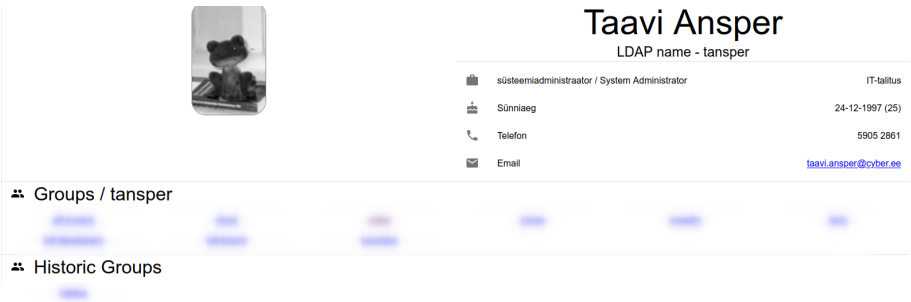
Osakondade pildivaade

- Kasutaja näeb keritavat ala, kus asetsevad kastid, mille sees on töötajate pildid.
- Töötajate piltide all on nende nimi.
- Töötajad, kes kuuluvad ühte osakonda on grupeeritud kokku.
- Pildikastide vahel on eraldusriba, kus on kirjas osakonna nimetus.
- Vajutades osakonna vaates kasutaja pildikasti peale, suunab rakendus edasi kasutajavaatesse.
- Osakondade pildivaade on kuvatud joonisel [8].

Joonis 8. *Osakondade pildivaade. Lõputöö autori andmed on näitena jäetud udustamata. Kasutajal pole veel uues ERP süsteemis pilti ülesse laetud. Selletõttu kuvatakse vaikepilti.*

Kasutajavaade

- See täidab nõuet nr.9 peatükist 3.1.1. Kasutajavaade on kuvatud joonisel [9].
- Kasutajavaates näeb kasutaja kohta järgmist infot:
  1. Nimi
  2. Meiliaadress
  3. Töötelefon
  4. Osakond, kus vaadatav isik töötab
  5. Ametikoht
  6. Vanus
  7. Kasutaja pilt
  8. LDAP grupid, kuhu kasutaja kuulub.



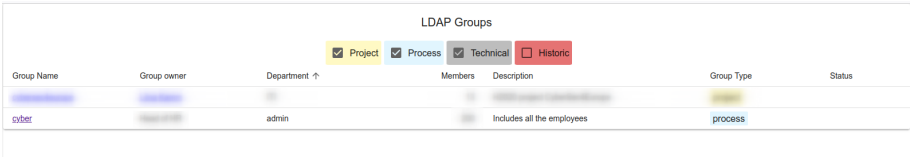
Joonis 9. *Kasutajavaade. Kuvatud on lõputöö autori andmed.*

LDAP gruppide vaade

- LDAP gruppide vaade on kuvatud joonisel [10].
- Kasutaja näeb tabelit LDAP gruppidest:
  1. Grupi nimi
  2. Grupi omanik
  3. Osakond, millele grupp kuulub

4. Grupi liikmete arv
5. Kirjeldus
6. Grupi tüüp
   - "*Process*"
   - "*Project*"
   - "*Technical*"
7. Grupi olek (arhiveeritud või aktiivne)

- Tabelit saab sorteerida kõigi veergude järgi vajutades veeru pealkirjale.
- Tabeli kohal on neli märke ruutu, mis vastavad gruppi tüübile, neid valides saab kasutaja filtreerida vastavaid grupitüüpe.



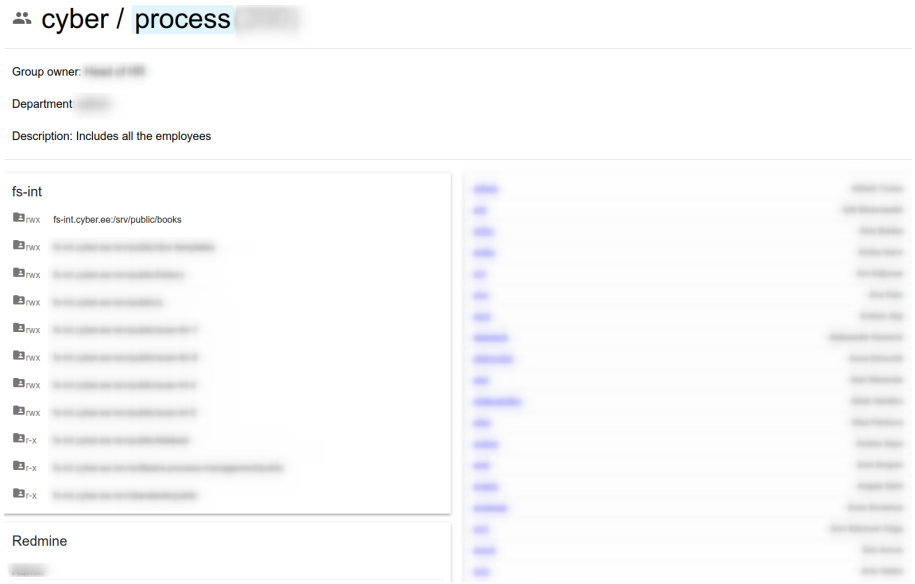
Joonis 10. *Gruppide vaade. Otsingulahtris märksõna 'cyber' kasutades leitud tulemused. Kuvatud peamist cyberi gruppi.*

LDAP grupi vaade

- LDAP grupi vaade on kuvatud joonisel [11].
- Kasutaja näeb LDAP grupi vaates sama infot, mida kuvatakse grupi kohta ka LDAP gruppide tabeli vaates.
- Lisaks näeb kasutaja ka kasutajate nimekirja, kes on selle grupi liikmed. Grupiliikme nimele peale vajutades suunatakse edasi kasutajavaatesse.
- Kasutaja näeb iga LDAP grupi pääsuõigusi järgnevates süsteemides:
  1. Redmine - loetelu projektidest, millele grupp ligi saab.
  2. Wiki - loetelu wiki nimeruumidest, millele grupp ligi saab.
  3. failiserver - loetelu kataloogidest, millele grupp ligi saab.

Puhkuste vaade

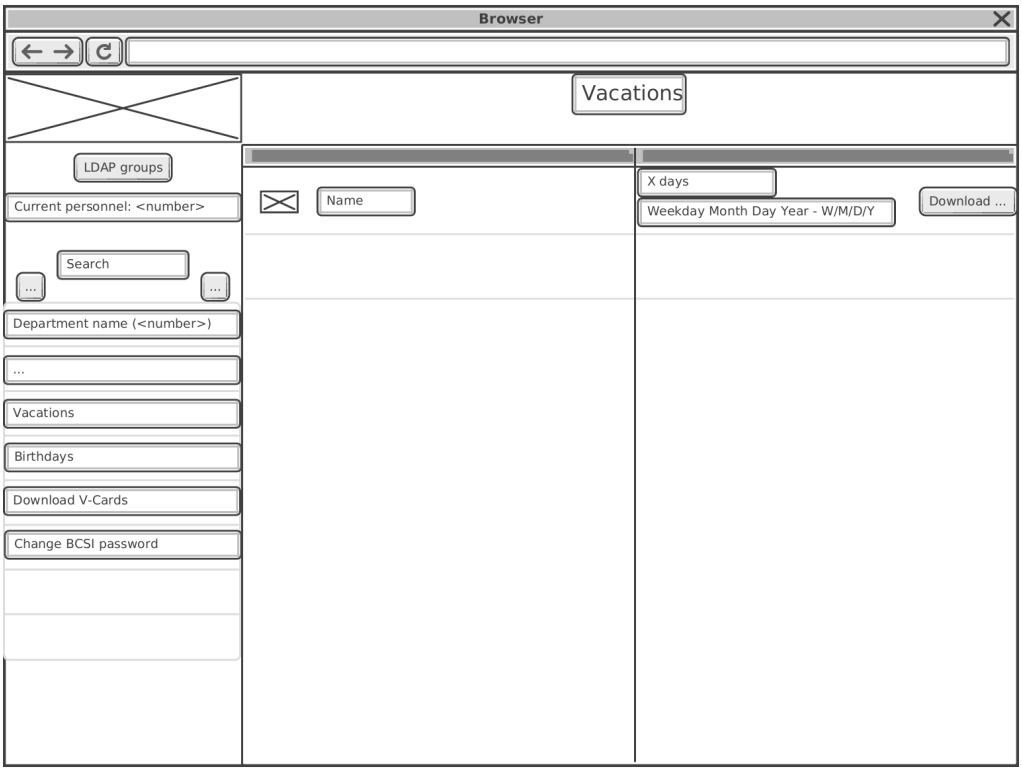
- Puhkuste vaate sõrestikmudel on kuvatud joonisel [12].
- Nimekiri praegustest ja tulevatest puhkustest.
- Iga puhkuse kirje juures on töötaja nimi ja väike avatar, kus on kasutaja foto.
- Iga puhkuse kirje juures on puhkuse algkuupäev ja lõppkuupäev ning nende vahemik päevades.
- Kirje lõpus on hüperlink, mis võimaldab alla tõmmata iCal tüüpi faili, mille sisu on antud puhkuse info.



Joonis 11. *Grupivaade. Kuvatakse peamist cyberi gruppi.*

Sünnipäevade vaade

- Sünnipäevade vaate sõrestikmudel on kuvatud joonisel [13].
- Nimekiri ettevõtte töötajate sünnipäevadest.
- Iga kuu on eraldatud vaheribaga ja kuu nimega.
- Töötaja juubel on tähistatud boldis tekstiga.



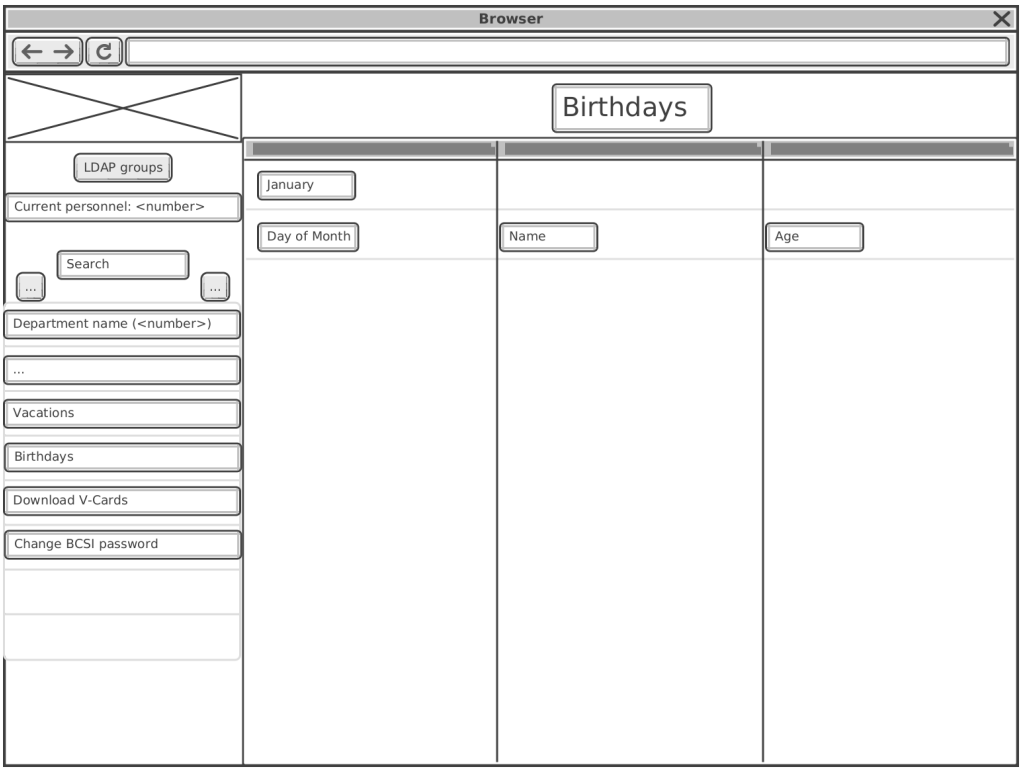
Joonis 12. *Puhkuste vaate sõrestikmudel*

4.2 ERP süsteemi kasutajakontode loomine ja haldus

Uus ERP süsteem lubab kõikidel kasutajatel sinna sisse logida, et hallata oma personaliinfot, esitada puhkusetaotlusi jms. Kasutajate andmebaas on ERP süsteemis lokaalne, seda ei saa üldise autentimislahendusega liidestada. Keskne kasutajate haldus toimub LDAP süsteemis. Kuna ERP süsteemi saab luua kasutajaid kasutades API-t, siis sünkroniseerib personaliinfo portaal kasutajaid LDAP-ist ERP süsteemi.

ERP süsteemi luuakse kõigile aktiivsetele kasutajatele konto, mille kasutajanimi on identne LDAP-i kasutajanimega. Sünkroniseerimisel kontrollib rakendus ka olemasolevaid ERP kontosid ja kui seal leidub kasutaja, mis pole enam LDAP-is aktiivne, siis see konto märgitakse suletuks või kustutatakse ERP süsteemist. Kasutajaid, kes on vähemalt korra ERP süsteemi sisse loginud, kustutada ei saa, neid saab ainult suletuks märkida.

Kasutajad luuakse algselt ilma paroolita ja ERP teenus ei luba ilma paroolita kasutajatel sisse logida. Selletõttu on vaja kasutajal moodust oma parooli seada ja vahetada. Paroolivahetus on personaliinfo portaali funktsioon.



Joonis 13. *Sünnipäevade vaate sõrestikmudel*

4.2.1 LDAP ja ERP sünkronisatsioon

Rakendus sünkroniseerib automaatselt LDAP-is kirjeldatud aktiivsed kasutajad ERP süsteemi. LDAP-i kasutajainfost saab rakendus info, mille põhjal ERP süsteemi uusi kasutajaid luua:

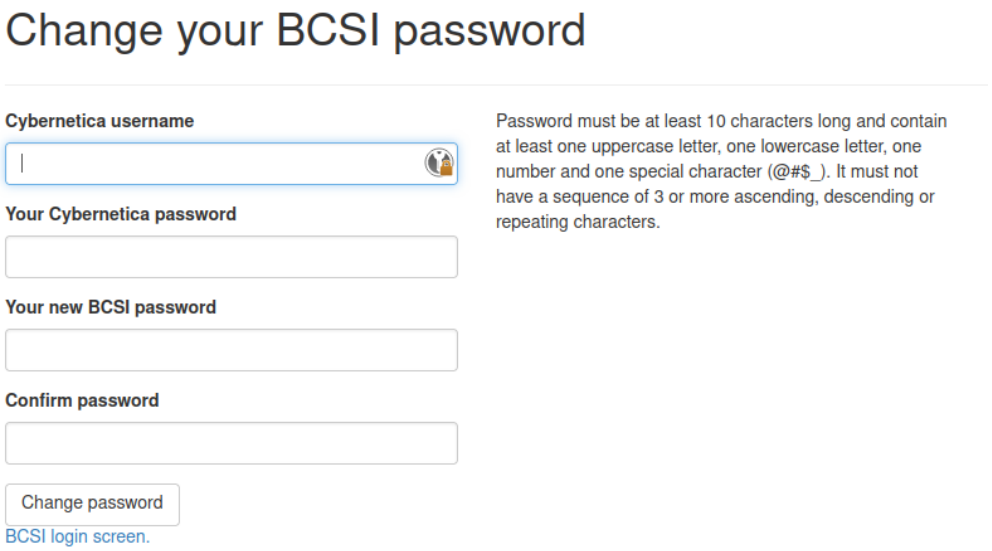
- UID
- email
- DisplayName

4.2.2 ERP konto parooli muutmine

Portaal võimaldab kasutajal vahetada enda parooli ERP süsteemis. Selleks peab kasutaja end personaliinfo portaalile autentima kasutades oma LDAP parooli. ERP teenuse kasutajate paroolivahetuse lehekülg on kuvatud joonisel [14]. Kasutaja sisestab järgmised andmed:

- LDAP kasutajanimi
- LDAP kasutaja parool
- Uus ERP süsteemi parool

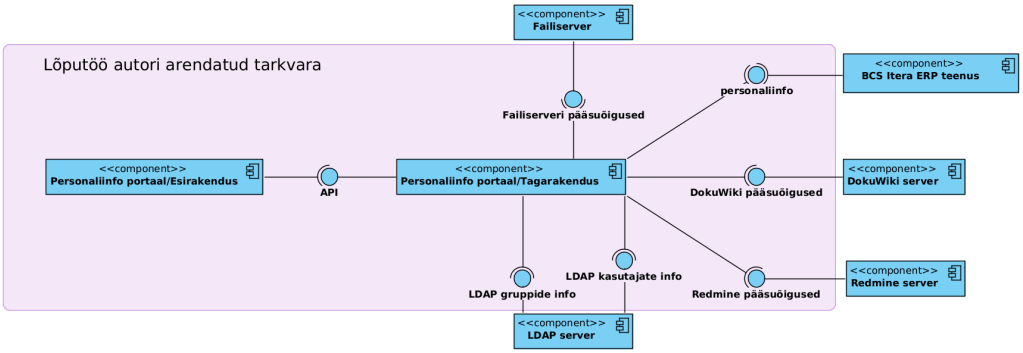
- Uue ERP süsteemi parooli kordus



Joonis 14. *ERP teenuse kasutajate paroolivahetuse lehekülg.*

5. Uue süsteemi arhitektuur

Rakendus jaguneb kaheks osaks: eesrakendus ja tagarakendus. Rakenduse kolmanda versiooni struktuur on kujutatud joonisel [15].



Joonis 15. *Personaliinfo portaali kolmanda versiooni struktuur.*

5.1 Tagarakendus

Tagarakenduse peamine funktsioon on olla vahekiht eesrakenduse ja kõikide süsteemide vahel, mille andmeid kuvatakse personaliinfo portaali eesrakenduses. Liidestatud süsteemid pakuvad tagarakendusele teenuseid, millele lõppkasutajad ei tohi ligipääsu omada. Tagarakendus filteerib saadud andmetest välja kõik tundlikud andmed, enne kui need eesrakendusele edastab.

Ühenduvus nende teenustega käib erineval viisil, kas kasutades veebiteenuseid või SSH-d.

5.1.1 Tagarakenduse liidestused

ERP Tagarakendus saab ligi ERP teenusele kasutades API-t. Tagarakendus peab end API-le autentima API võtme abil. Suhtlus on turvatud HTTPS protokolli abil. ERP rakendusest päritakse töötajate personaliandmeid. ERP API annab juurdepääsu kõigile ERP süsteemi funktsioonidele. Päritud andmete hulgas on palju andmeid, mida kasutajad ei tohi eesrakenduse kaudu näha. Sellepärast filtreeritakse ebavajalikud andmed tagarakenduse poolt välja.

LDAP Tagarakendus saab ligi LDAP serverile kasutades LDAP-i protokolli. Kasutusel olev sisemine LDAP server lubab anonüümset ühendumist, ehk kõik, kes saavad sisevõrku näevad sealseid andmeid: kasutajaid ja gruppe. Eesrakendus saaks ühenduda LDAP serveriga ka otse, kasutades NodeJS ldap-i moodulit, aga kuna andmeid on vaja enne töötleda oli parem teha liidestus läbi tagarakenduse. Ühendumiseks LDAP serveriga kasutab rakendus teeki python-ldap[12]. Tagarakendus töötleb LDAP-i serverist tulnud andmed JSON formaati ja neid andmeid saab pärida tagarakenduse API kaudu. Sealhulgas on oluline osa LDAP grupi kirjeldusväli, mis on jaotatud semikoolonitega eri osadeks, millest tagarakendus loeb välja grupi atribuudid.

Redmine Tagarakendus saab ligi töökäskude halduse süsteemile Redmine kasutades API-t ja talle ette antud API võtit, millega tagarakendus saab pärida andmeid üle HTTPS protokolli. Päringute tegemiseks on kasutusel Pythoni teek python-redmine[13]. Turvalisuse pärast on tehtud eraldi Redmine-i pseudokasutaja nimega bot-redmine, kes on lisatud kõikide Redmine projektide liikmeks. Sellel kasutajal on piiratud õigused ja ta näeb ainult kasutajaid ja gruppe, kes on projektide liikmed, muid õigusi tal projektides pole.

Wiki DokuWiki pääsuõigused asuvad failis acl.auth.php, mis on DokuWiki töötava eksemplari failipuus. Personaliinfo portaal jookseb teises serveris. Turvalisuse huvides kopeeritakse pääsuõiguste fail personaliinfo portaali serverisse kahe sammuga. Pääsuõiguste fail kopeeritakse Wiki serveris cron utiliidi abil iga päev kindlal kellajal eraldi kausta, millele on lugemisõigus Linuxi kasutajal bot-personal. See kasutaja ei saa serverist lugeda muid faile. Selle kasutaja SSH lubatud võtmete hulka on lisatud personaliinfo portaali SSH avalik võti. Kasutades Linuxi utiliiti scp kopeerib personaliinfo portaal enda failipuusse DokuWiki serverist selle pääsuõiguste faili, millest tagarakendus loob JSON formaadis andmed, mida API otspunktist saab pärida eesrakendus.

Failiserver Failiserveri pääsuõiguste info lugemine on tehtud sarnaselt DokuWiki lahendusele. Erinevus seisneb selles, et pääsuõiguste fail luuakse kasutades Linuxi utiliiti getfacl, mis annab failiserveris olevate kaustade juurdepääsu õigused.

5.1.2 Tagarakenduse teenused

API Tagarakendus pakub API teenust otspunktil '*https://example-url.com/api/v1/*'. Pärast api versiooni sisestamist tulevad erinevad teenused, mida tagarakendus pakub (edaspidi URL-i esimest poolt ei korrata.). ERP teenusest tulevad andmed, mida tagarakendus töötleb on kättesaadaval otspunktil '*/bcsi/employees*', mis annab kõikide töötajate kohta infot baasinfot. Lisaks on ettevõtte struktuuri kätte saamiseks otspunkt '*/bcsi/departments*'. Töötajate pilte saab kätte otspunktil '*/bcsi/pictures/<töötajanimi>*', kus nurksulgude vahel olev osa peab asendama ettevõtte töötaja nimega.

ERP teenuse paroolivahetus veebiliides Rakendus lubab ka vahetada ettevõtte töötajatel enda ERP teenuse kasutajakonto parooli. See asub otspunktil '*https://example-url.com/utils/change-bcsi-password*', ning kuvab veebivormi, kus kasutaja saab sisestada andmed paroolivahetuseks ERP teenuses. Kasutaja sisestatud LDAP-i andmeid kasutatakse, et luua ühendus antud kasutajana LDAP serverisse, kui ühendumine on edukas, on kasutaja autenditud ja tal lubatakse muuta ERP rakenduse parooli.

5.2 Eesrakendus

Eesrakendus kasutab standarseid Reacti arenduses kasutatavaid NodeJS mooduleid, et uuendamine oleks tulevikus võimalikult lihtne. Need on:

- react-router-dom
- react-dom
- react-scripts
- mui/material - veebikomponentide kujundus Material UI.[14]

5.3 Realiseermise käigus tehtud muudatused

Rakenduse realiseerimise käigus tulid välja mõned probleemid, mis nõudsid muudatusi rakenduse arhitektuuris, mida algselt ei olnud ette nähtud.

Vahemälu Rakendusele tuli juurde arendada vahemälu, kuna muidu poleks rakenduse kasutuskiirus olnud mõistlik. Põhiline mure seisnes ERP rakenduse päringutes.

ERP rakendusse saab iga kasutaja laadida oma profiilile ülesse oma foto. Pärides API kaudu ERP rakenduselt kasutajate andmeid, saadetakse see pilt ka kaasa. Testkeskkonnas, kus kasutajatel pilti polnud, olid API päringud väiksed - kilobaitides mõõdetavad andmemahud. Koos piltidega olid andmete mahud megabaitides ja selle andmehulga laadimine võtab kümneid sekundeid. Selletõttu ei saa otse läbi tagarakenduse pärida ERP rakenduselt andmeid.

Sarnaselt kasutajate sünkronisatsioonile, käib rakenduse küljes taimer, mis kutsub välja funktsiooni, mis uuendab neid andmeid, tehes ERP teenuse pihta perioodiliselt API päringuid ja salvestades andmed failisüsteemi. Kuna pildid on väga mahukad, siis on mõistlik nad muudest andmetest eraldada. Funktsioon, mis salvestab andmed failipuusse, eraldab andmete küljest pildid ja salvestab andmed ja pildid eraldi. Nii saab eesrakendus andmeid pärida kiiresti.

Pilte on vaja ikkagi kuvada, seetõttu pildid salvestatakse kettale eraldi muudest andmetest. Selle jaoks on olemas teine funktsioon, mis otsib salvestatud andmetest, kus on olemas pildid, ülesse pildid ja salvestab need vastava kasutaja nimega kettale vahemällu stiilis: 'EesnimiPerenimi'. Faili laiendit ei ole lisatud, kuna pilt ei pruugi olla ainult JPEG formaadis. Rakenduse sisse ehitatud taimer kutsub välja ka seda funktsiooni, mis uuendab pilte vastavalt vajadusele. Kui kasutajal on juba pilt salvestatud failipuusse, siis enne uuendamist võrreldakse selle kontrollsummat uue, andmetest tulnud pildi kontrollsummaga, kui vana erineb uuest, siis kirjutatakse fail üle. Tagarakenduse API kaudu saab neid pilte pärida. Otspunkt kasutab otsimiseks EesnimiPerenimi vormi. nt. /api/v1/pictures/EesnimiPerenimi .

6. Uue süsteemi ehitamine ja paigaldus/evitamine

Rakenduse lähtekoodi haldab Cybernetica sisemine versioonihaldussüsteem Gitlab. Seadistatud on automaatne ehitusprotsess, mis loob Dockeri[15] kettapildi. Automaatset ehitusprotsessi haldav kood on lisatud tööle juurde Lisana [7].

Personaliinfo portaal töötab eraldi Debian Linux virtuaalmasinas. Personaliinfo portaal evitatakse sinna virtuaalmasinasse kasutades Ansible tarkvara [16].

6.1 Ehitamine

Rakenduse uue versiooni ehitamiseks on vaja teha koodimuudatus ja see Gitlabi üles laadida kasutades Giti. Rakenduse lähtekoodis on fail '.gitlab-ci.yml' [7]. Selles failis on defineeritud, kuidas rakendust ehitada.

Gitlabi CI fail jaguneb mitmeks osaks:

- Ehitusprotsess, mis jaguneb omakorda:
  - eesrakenduse ehitus;
  - tagarakenduse ehitus.
- Eduka ehitusprotsessi tehiste evitamine.

Ehitusprotsess Ehitusprotsess kasutab lähtekoodis asuvat 'Dockerfile'-i[17], milles on defineeritud sammud, kuidas ehitada rakenduse kettapilt. Nii esi, kui tagarakendusel on oma 'Dockerfile' ja ehitusprotsessi lõpptulemuseks on kaks eraldi kettapilti, üks sisaldab eesrakendust ja teine tagarakendust.

Tehiste evitamine Pärast edukat ehitusprotsessi on võimalik versioneerida selle ehitusprotsessi tehised. Selle jaoks on vaja kasutada Giti tööriista funktsiooni 'git tag', mille tulemusena lisatakse tehistele versiooninumber.

6.2 Evitamine

Pärast edukat automaatset ehitusprotsessi on Gitlab-i registris olemas uus Dockeri kettapilt. Enne kui sellest kettapildist saab luua töötava Dockeri konteineri on vaja seadistada virtuaalmasin, kus personaliinfo portaal jookseb. Selleks otstarbeks on kasutusel Ansible tarkvara.

Debiani virtuaalmasin läbib kõigepealt üldise seadistamise, mis tehakse kõikide virtuaalmasinate puhul ettevõtte infrastruktuuris. See seadistus hõlmab erinevaid väikeseid asju. DNS serverite määramine, automaatsete uuenduste seadistamine, meiliserverite seadistus jms. Pärast üldist seadistust lisatakse Ansible abil spetsiifilisem seadistus. Personaliinfo portaali puhul seadistatakse virtuaalmasinasse Dockeri tööriistad. Kopeeritakse Dockeri seadistusfail 'docker-compose.yml', mis kasutab Dockeri lisamoodulit 'Docker Compose'[18], mis aitab hõlpsamalt Dockeri rakendusi käitada.

Personaliinfo portaali docker-compose.yml fail [7] koosneb koosneb kolmest osast:

1. eesrakenduse Dockeri konteiner;
2. tagarakenduse Dockeri konteiner;
3. pöördproksi Dockeri konteiner.

Igale Dockeri konteinerile saab juurde anda ka keskkonnamuutujad, läbi mille abil jagatakse rakendusele sätteid. Läbi nende keskkonna muutujate on antud Tagarakendusele erinevad saladused, millega saab ligi personaliinfo portaali liidestatud rakendustele. Nt. ERP teenuse API võti. Redmine-i teenuse API võtmed ja ka SSH privaatvõtme fail ja selle privaatvõtme salasõna.

Rakenduse avamiseks kasutajatele on kasutusel Docker konteineris jooksev pöördproksi, mis kasutab ka OWASP sihtasutuse loodud veebitulemüüri *modsecurity-crs-docker* [19], mis kasutab pöördproksina laialdaselt levinud Nginx-i[1] ja veebitulemüüri moodulit ModSecurity Core Rule Set[20] ehk *WAF*-i.

Ansible konfiguratsioon asub Gitlabis ja sellele saavad ligi ainult IT osakonna töötajad ja muud olulised isikud. Need eelmainitud saladused, mis on vajalikud tagarakenduse tööks, asuvad ka seal Ansible Giti hoidlas. Hoolimata, et seda hoidlat näeb väike hulk inimesi, on kõik sealsed saladused krüpteeritud kasutades ansible-vault[21] tarkvara, mis kasutab saladuste krüpteerimiseks AES256 standardit.

[1] https://www.nginx.com/

Viimane oluline lüli rakenduse töös on Linuxi süsteemse tarkvara 'systemd' teenusefail. Systemd on tarkvara, mis vastutab selle eest, et teenused töötaksid ja nt. taaskäivtamiste puhul tuleksid uuesti ülesse. Loodud systemd teenuse fail kontrollib, et 'docker-compose.yml' failis etteantud seadistus oleks aktiivne ja rakendus töötaks.

7. Kokkuvõte

Töö käsitleb personaliinfo portaali arendust ettevõtte Cybernetica AS näitel. Kirjeldatakse eelnevaid versioone personaliinfo portaalist ja nende puudusi. Portaali arenduse sisendiks oli erinevate kasutajate tagasiside, mis saadi töövoohaldus keskkonda kirjutatud kommentaaridest, kasutajatega arutades ja ka vaadeldes olemasolevaid andmeid, mis funktsionaalsust nendega saab luua. Oluline oli tuua üle juba olemasolev funktsionaalsus, mis oli olemas rakenduse eelmistes versioonides.

Töös käsitletava rakenduse kolmanda versiooni arenduse peamine põhjus oli andmete algallika vahetus, loobuti eelmisest Fujitsu Persona teenusest, et kasutada uut ERP teenust, mida pakub BCS Itera. Selletõttu muutus andmete vorm ja ka osaliselt andmete sisu, mida pärida saab. Uus ERP teenus lubab kasutajatel sisselogida enda andmetele ligipääsemiseks ja puhkustetaotluste esitamiseks. Antud teenus kasutab lokaalseid kasutajaid ja selletõttu pidi personalirakendus sünkroniseerima olemasolevad LDAP-i kasutajad ERP teenuse kasutajate andmebaasiga, kuna ERP rakendusel puudus parooli seadistamise võimalus, pidi looma ka veebilehe, kus kasutaja saab enda ERP teenuse kasutaja parooli muuta.

Realisatsiooni käigus tuli välja probleem andmete suure mahuga API päringutes ja selletõttu tuli portaali tagarakenduses kasutada vahemälu, et päringud oleks kiiremad ja kasutajatel oleks mugav portaali kasutada. Rakenduse mittefunktsionaalsed nõuded sai täidetud ja funktsionaalsetest nõuetest jäi täitmata ainult puhkuste kuvamine, kuna personaliosakond ei olnud uut ERP rakendust täielikult kasutusel võtnud ja puudusid puhkuseandmed, mida kasutada.

Rakenduse paremaks arendamiseks võeti kasutusele Gitlab-i sisseehitatud CI/CD võimekus, mille abil saab hõlpsalt ehitada uusi versiooni rakendusest, kui on lähtekoodis muutusi. Rakendus pakendatakse kahte eraldi Dockeri kettapilti - eesrakenduse kettapilt ja tagarakenduse kettapilt. Rakendus evitati Debian Linuxi virtuaalmasinasse, kus ta jookseb Dockeri konteineritena. Rakenduse ees on kasutusel ka pöördproxy, mis kasutab veebitulemüüri.

Edasine plaan rakenduse arenduses on kasutajatega testimine ja nende tagasiside kaudu erinevate vigade parandus. Lisaks on planeeritud ka tulevikku läbistustest kasutades OWASP ASVS raamistikku[22].

Kasutatud kirjandus

[1] LLC CSF Tools. *STRIDE-LM Threat Model*. URL: https://csf.tools/reference/stride-lm/.

[2] Altexsoft. *Non-functional Requirements: Examples, Types, How to Approach*. URL: https://www.altexsoft.com/blog/non-functional-requirements/.

[3] Microsoft Dynamics BC partner BCS Itera AS. *Palk 365 ja Personal 365 liidese ulatuse dokumentatsioon*. 2022.

[4] Python Software Foundation. *Python*. URL: https://docs.python.org/3/.

[5] Armin Ronacher. *Flask Framework*. URL: https://flask.palletsprojects.com/en/2.2.x/.

[6] Miguel Grinberg. *Flasky*. 2018. URL: https://www.flaskbook.com.

[7] Microsoft. *Typescript*. URL: https://www.typescriptlang.org/docs.

[8] Inc Meta Platforms. *ReactJS*. URL: https://reactjs.org/docs/getting-started.html.

[9] Pyenv org. *Pyenv*. URL: https://github.com/pyenv/pyenv.

[10] *Poetry*. URL: https://python-poetry.org/.

[11] *NVM*. URL: https://github.com/nvm-sh/nvm.

[12] *python-ldap*. URL: https://python-ldap.org/en/latest.

[13] Maxim Tepkeev. *python-redmine*. URL: https://python-redmine.com/.

[14] *Mui*. URL: https://mui.com.

[15] Docker Inc. *Docker*. URL: https://docs.docker.com/.

[16] Inc Red Hat. *Ansible*. URL: https://docs.ansible.com/ansible/latest/index.html.

[17] Docker Inc. *Dockerfile*. URL: https://docs.docker.com/engine/reference/builder/.

[18] Docker Inc. *Docker Compose*. URL: https://docs.docker.com/compose/.

[19] Inc OWASP Foundation. *ModSecurity Core Rule Set Docker Image*. URL: https://github.com/coreruleset/modsecurity-crs-docker.

[20] OWASP Foundation. *ModSecurity Core Rule Set*. URL: https://coreruleset.org/.

[21] Inc Red Hat. *Ansible Vault*. URL: https://docs.ansible.com/ansible/latest/vault_guide/index.html.

[22] Inc OWASP Foundation. *OWASP Application Verification Standard*. URL: https://owasp.org/www-project-application-security-verification-standard/.

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Taavi Ansper

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose "Ettevõtte personaliinfo portaali arendus", mille juhendajad on Toomas Lepik ja Tarmo Oja .
   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.

06.01.2023

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

Lisa 2 – Docker Compose Fail

Personaliinfo portaali docker-compose.yml fail

```
---
version: "3.8"
services:
  nginx-modsecurity-proxy:
    container_name: nginx-modsecurity-proxy
    hostname: nginx-modsecurity-proxy
    image: owasp/modsecurity-crs:nginx
    restart: unless-stopped
    volumes:
        - /opt/personal/nginx_modsecurity/server.crt:/etc/nginx/conf/server.crt:ro
        - /opt/personal/nginx_modsecurity/server.key:/etc/nginx/conf/server.key:ro
        - /opt/personal/nginx_modsecurity/log:/var/log/nginx
        - /opt/personal/nginx_modsecurity/REQUEST-900-EXCLUSION-RULES-BEFORE-CRS.conf:/etc/modsecu
        - /opt/personal/nginx_modsecurity/default.conf.template:/etc/nginx/templates/conf.d/defaul
    ports:
        - "80:80"
        - "443:443"
    env_file:
      - '/opt/personal/nginx_modsecurity/env-nginx_modsecurity'
    networks:
      - personal-network

  personal_backend:
    container_name: personal_backend
    hostname: personal_backend
    image: gitlab.cyber.ee:5050/REDACTED/personal/personal-api:v0.3.3
    restart: unless-stopped
    volumes:
      - /opt/personal/personal_backend/data:/var/cache/personal
    env_file:
      - '/opt/personal/personal_backend/env-personal-backend-config'
    networks:
      - personal-network
  personal-frontend:
    container_name: personal-frontend
    hostname: personal-frontend
    image: gitlab.cyber.ee:5050/REDACTED/personal/personal-page:v0.3.3
    restart: unless-stopped
    env_file:
```

```
      - '/opt/personal/personal_frontend/env-personal-frontend-config'
    networks:
      - personal-network
networks:
  personal-network:
    driver: bridge
```

Lisa 3 – .gitlab-ci.yml

Personaliinfo portaali .gitlab-ci.yml fail.

```
---
stages: # List of stages for jobs, and their order of execution
  - build
  - release

before_script:
  - docker login -u $CI_REGISTRY_USER -p $CI_REGISTRY_PASSWORD $CI_REGISTRY
  - chmod +x ./setup_env.sh
  - . "./setup_env.sh"

.build: # This job runs in the build stage, which runs first.
  stage: build
  script:
    - |
      IMAGE_COMMIT_REF=$CI_REGISTRY_IMAGE/$CONTAINER_NAME:latest
      docker build \
        --pull \
        --tag $IMAGE_COMMIT_REF \
        --file $CONTAINER_NAME/Dockerfile \
        $CONTAINER_NAME
    - docker push $IMAGE_COMMIT_REF

build node app:
  extends: .build
  variables:
    CONTAINER_NAME: 'personal-page'

build python api:
  extends: .build
  variables:
    CONTAINER_NAME: 'personal-api'

## This job creates docker release image
.release-image:
  rules:
    - if: '$CI_COMMIT_TAG != null'
  stage: release
  variables:
    CONTAINER_TAGGED_IMAGED: '$CI_REGISTRY_IMAGE/$CONTAINER_NAME:$CI_COMMIT_TAG'
```

```
    IMAGE_COMMIT_REF: '$CI_REGISTRY_IMAGE/$CONTAINER_NAME:latest'
  script:
    - docker pull "$IMAGE_COMMIT_REF"
    - docker tag "$IMAGE_COMMIT_REF" "$CONTAINER_TAGGED_IMAGED"
    - docker push "$CONTAINER_TAGGED_IMAGED"

release node app:
  extends: .release-image
  variables:
    CONTAINER_NAME: 'personal-page'

release python api:
  extends: .release-image
  variables:
    CONTAINER_NAME: 'personal-api'
```