

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Andreas Poola
Rakenduse “NobeTehas” arendamine
Bakalaureusetöö (9 EAP)

Juhendaja: M.Sc Vambola Leping

Tartu 2022

Rakenduse “NobeTehas” arendamine

Lühikokkuvõte:

Bakalaureusetöö eesmärgiks on arendada front-end rakendus, mis on suureks abiks tehase tööliste nende igapäevaste tööde majandamisel ja aja märkimisl. Rakendus on üles ehitatud klient-serveri arhitektuuril. Serveri poolel on kasutatud NestJS ja GraphQLi. Kliendi poolel on kasutuses Angular ja Ionic raamistikud. Bakalaureusetöö annab ülevaate kliendist ja firmast kes tarnis rakenduse, kliendi esitatud nõuetest uuele rakendusele, rakenduse valmimisest, Angulari ja Ionicu raamistiku kasutamine ja valmis saanud rakendusest.

Võtmesõnad: Rakenduse loomine, Angular, Ionic

CERCS: P175 Informaatika

Developing of application “NobeTehas”

Abstract:

The aim of the bachelor’s thesis is to develop a front-end application, which is of big help to the factory workers for managing their everyday tasks and to count time spent on said tasks. The application is built on client-server method. The server side is built with NestJS and GraphQL. Client side is built with Angular and Ionic frameworks. The thesis gives an overview of the client and the company that developed the application, the requirements for the new application, development of the application, the use of Angular and Ionic frameworks and the final frontend of the application.

Keywords: Development of an application, Angular, Ionic

CERCS: P175 Informatics

Sissejuhatus	4
1 Mõisted ja terminid	5
2 Tutvustus firmadest.....	7
2.1 Klient: <i>Plaat Detail</i>	7
2.2 Mobiilirakenduse looja: <i>Inspirators! OÜ</i>	7
3 Kliendi esitatud nõuded uuele mobiilirakendusele	9
3.1 Lisatööd.....	9
3.1 Kasutajalood	11
3.1.1 Funktsionaalsed nõuded.....	11
3.1.2 Mittefunktsionaalsed nõuded	11
4 Kasutatud tehnoloogiad.....	12
4.1 Vana rakendus.....	12
4.2 Uus rakendus.....	12
5 Valminud rakendus	15
5.1 Teenused.....	15
5.1.1 Ülesannete kättesaamise teenus	15
5.1.2 Turvalisuse teenus.....	15
5.1.3 Strip teenus	15
5.2 <i>Front end komponendid</i>	15
5.2.1 Rakenduse komponent	15
5.2.2 Sisselogimise komponent.....	16
5.2.3 Menüü komponent	16
5.2.4 Ülesande komponent.....	17
5.2.5 Ülesande nimekirja komponent	18
5.2.6 Ülesannete kuvamise komponent.....	19
5.2.7 Hetke käimas olevate ülesannete komponent.....	20
5.2.8 Plaani komponent.....	21
5.2.9 Ülesannete jaotamise komponent.....	21

6 Kokkuvõte.....	23
Kasutatud kirjandus	24
Lisad	25
<i>Litsents</i>	<i>25</i>

Sissejuhatus

Bakalaureusetöö eesmärk on luua front end rakendus firmale Plaat Detail. Aastal 2016 soovis Plaat Detail endale tööde majandamise rakendust, et kaotada paberi kujul töökäsud. Algne firma, kellelt sooviti sellist rakendust oli ProExpert, kes selle ka tarnis. Tööde maht oli alahinnatud ning ka kliendi sisend kogu arenduse jooksul ei olnud piisavalt kvaliteetne. Sellegi poolest valmis rakendus, mis oli kasutuses tehastes 4-5 aastat.

Aastal 2021 tuli Plaat Detail IT-idufirma Inspirators!i juurde murega, et kogu lahenduse infrastruktuur ei olnud piisavalt töökindel, mis põhjustas katkestusi rakenduse töös. See omakorda põhjustas katkestusi tehase töös. Sellepärast oli vaja uuendada nii back- kui front-end komponente. Selle 4-5 aastaga sai Plaat Detail aru, kui väga neil sellist rakendust ka vaja läheb, ning ei tahtud sellest lahendusest loobuda. Aastal 2021 algaski koostöö Inspirators!iga ja rakenduse uuendamine.

Rakendust ise oli vaja selleks, et eemaldada paberimajandust tehases. Tehase töölised näevad rakenduses ülesandeid mis on vajavad tegemist, mis on pooleli jäänud, uued, lõpetatud ülesanded ja tähtsad. Lisaks loeb rakendus aega, kui kaua on tööline tegelenud mingite ülesannetega. See info saadetakse kõik tagasi andmebaasi ja sealt näeb juba töödejuhataja, mis ülesanded on valmis ja kaua nende jaoks aega on kulunud.

Bakalaureusetöö sisu on jaotatud 5 põhiliseks peatükiks. Esimeses peatükis selgitatakse töös kasutuses olevad erinevad mõisted ja terminid. Teises peatükis on tutvustus kliendist Plaat detailist ja mobiilirakenduse looja firmast Inspirators!. Kolmandas peatükis on ülevaade kliendi esitatud nõuetest uuele mobiilirakendusele ja kasutaja lood. Neljandas peatükis on välja toodud kasutatud tehnoloogiad uue rakenduse loomiseks. Viimane peatükk on kokkuvõte, kus kirjeldatakse lühidalt tehtut.

1 Mõisted ja terminid

Front-end: kliendipoolne vaade[1]

Back-end: kasutajale nähtamatu tagasüsteem[1]

Industry 4.0: tööstus 4.0 ehk targad tehased, kus on kaastatud masinõpet[2]

SAF: eestimaine finants- ja majandustarkvara[3]

REST: esitusoleku siire, harus töötluste paradigma[1]

JWT: JSON-i veebimärk, kasutatakse digitaalse allkirja abil kontrollitava ja usaldusväärse teabe edastamiseks.[4]

Teek: infoobjektide kogu[1]

SDK: tarkvaraarendus komplekt[1]

API: Rakendusliides[1]

DDD: Ehk Domeenipõhine disain on tarkvara kujundamise lähenemisviis, mis keskendub tarkvara modelleerimisele, et sobitada domeeni vastavalt selle domeeni ekspertide sisendile[5]

NDK: Võrguarenduskoplekt [6]

CQRS: Command and Query Responsibility Segregation(Käsu ja päringu vastutuse eraldamine), on rakenduse arhitektuuri muster.[7]

ES: Sündmusehõive, ehk rakenduse muudatuste salvestamine sündmusejadana [1]

LAGOM: Avatud lähtekoodiga raamistik Java ja Scala programmeerimis keeltele[8]

Storybook: Avatud lähtekoodiga tööriist, mis on kasutajaliidese ehitamiseks isolatsioonis[9]

2 Tutvustus firmadest

2.1 Klient: Plaat Detail

Plaat Detaili [10] kodulehe andmetele tuginedes on Plaat Detail 1996. aastal asutatud ettevõte. Põhiliselt tegeleb Plaat Detail laminaadi müügiga, kuid tegeleb ka laminaadist toodete valmistamisega. Plaat Detaili põhilised kliendid on mööblitootjad, sisustus- ja ehitusettevõtted ning ehitusmaterjalid kauplused. Plaat Detaili kolm põhisuunda on jae- ja hulgimüük, masstootmine ja eritootmine. 2011.a. avatud laminaat detaile tootev tehas on üks Põhjamaade suurimaid. Plaat Detail toodab 70000 ruutmeetrit laminaatdetaile, millest 90% eksporditakse välisurule.

2.2 Mobiilirakenduse looja: Inspirators! OÜ

Inspirators! on 06.05.2021 asutatud IT-idufirma, mille fookus on Industry 4.0 lahenduste loomisel[11]. Ettevõtte loodi, kuna Eestis on palju tootmis- ja ehitusettevõtteid, mis vajavad tarkvara, mis looks äriäärtust. Inspirators! on loonud reaalaraja andmeanalüütika lahenduse Eesti Gaasile, samuti IOT põhise kauglugemislahenduse Advenile. Inspirators!-i üks kõige kõrgema lisandväärtusega toode on FutuDraw, mis tänu oma innovaatsusele ja automaatikale võimaldab soojusarvestust ja ehitusjooniseid luua 20 000% kiiremini traditsioonilise mudeliga võrreldes. PlaatDetail kasutab Inspirators!-i mobiilirakendust igapäevases tootmisprotsessis, mille tulemusena on ettevõttel ülevaade tööaja arvestusest kui ka omahinnast iga tellimuse lõikes. Samuti võimaldab Inspirators! mobiilirakendus saata kasutajatele olulisi teavitusi masinate, tööolukordade jms kohta. PlaatDetaili mobiilirakendus on asendamatu töövahend tellimuste täitmisel.

Inspirators!-i pikaajaline fookus on jõuda tarkade toodete ja teenustega lisaks Eestile ka välisurule. Inspirators!-i lähenemine, luua esimese 5 päeva jooksul äriäärtust, on unikaalne ja esitab kõrgeid nõudmisi kõigile Inspirators! arendajatele.

Inspiraators! põhiline metoodika on hoida lähtekood kerge, testida tihti ja toimetada funktsionaalseid rakenduse tükke kohe, kui need valmis on.

Inspiraators! sai alguse kolmest inovaatorist, kelleks on Allan Poola, Allar Allas ja Kristi Pool. Allan Poola on tuntud kui üks Proeksperdi asutajaliige [12] ja pikaajaline Eesti e-riigi arengule panustaja. Allar Allas on tuntud kui uuenduslike IT-lahenduste ning kontseptsioonide arhitekt ja looja ning rahvusvaheliste projektide kogemustega agilne koots. Kristi Pool on tegelenud äriarendusega globaalses mastaabis - USA börsiettevõtetest Jaapani ja Kanada ettevõtetereni. Samuti on Kristi Poolil pikaajaline kogemus tarkvara valdkonna äri juhtimise ja partnerluse korraldamisega Skandinaavias. Kristi Pool on olnud ka Eesti e-riigi lahenduste kõneisik. Allan Poola on küll praeguseks ettevõttest lahkunud, samuti ka autor.

3 Kliendi esitatud nõuded uuele mobiilirakendusele

Kliendi nõuded mobiilirakendusele ei olnud algselt eriti suured. Põhiliseks nõudeks oli vana mobiilirakenduse ajakohastamine uue tehnoloogiaga. Ajakohastamist vajasisid mobiilirakendus, andmeliides, liides SAF'iga ja administreerimise veebirakendus. Üheks ajakohastamise eesmärgiks oli luua ka parem tarnemehhanism selleks, et uus funktsionaalsus liiguks edaspidi kiiremini ja mugavamalt kasutusse.

Pakkumise käigus lisandus uusi töid, mis tehti lisaks ajakohastamisele.

3.1 Lisatööd

- Tööd sorteerimine tähtpäevade järgi. Hetkel sorteeritakse töid SP(Suur paala, ehk tehase asukoht) numbri järgi, kui Klient soovis sorteerimist teostada tähtaja järgi
- Tööde prioriteeti peab saama individuaalselt muuta töölise kaupa
 - mobiilirakenduses
- Tööd kuvatakse sellises järjekorras, nagu serverist tuleb
 - Järjekorda saab korrigeerida admin liideses:
 - Grupipõhiselt (töölepingi järgi)
 - Isikupõhiselt
- Töö prioriteetsust saab muuta lohistades tööd nimekirjas ettepoole
 - Kui SAF-ist tulevad tööd peale, siis:
 - Kõige esimesena kuvatakse nimekirjas tööd, mis on meistri poolt varem juba nimekirja tipu viidud. Visuaalselt esile tõstetud (boldina)
 - Sealt edasi tulevad tööd tavapärases, tellimuse tähtajal põhinedes
- Tööde parandamine admin liideses:
 - Praegu eristatakse muutmisel tellimustöid ja muid töid, mis teeb administreerimise ebamugavaks
 - Tellimused ja muud tööd nii tööde lisamise kui nimekirjast muutmisel on vaja viia ühe ja sama universaalse vormi peale
- Andmete eksport PowerBi keskkonda.

- Hetkel koostab tootmisjuht exceli tootmisaegade andmetega käsitsi ning laeb ka PowerBi keskkonda käsitsi. Kuigi võimalik on andmed laadida ka ilma tootmisjuhi sekkumiseta PowerBi'sse ja andmete kogus on ka suurem kui käsitsi laadides.
- Probleemide kuvamine tootmisjuhile veebirakenduses
 - Tootmisjuht tahab operatiivselt saada ülevaadet tehases olevatest probleemidest. Kui töömehel tekib tootmises probleem, saab ta sellest teada anda rakenduse seest saadetud tekstisõnumiga. Tootmisjuht näeb talle saadetud sõnumeid administreerimise veebirakenduses
- Detailide arvu üle arvepidamine (ei ole veel implementeeritud) Rakenduses võiks näha kui töö käima panna, siis palju detaile tuleb töötajal teha.
 - Numbrid on SAFis saadaval
 - Kogused on detailipõhised
 - Ühel tellimusel võib olla mitu detaili
 - Jälgima peab samuti detailipõhiselt
- Võimalus kasutada tahvleid või suuremaid ekraane
 - Tootmisjuht tahab näha päeva- ja töötaja põhiselt ülevaadet tehtud töödest.
 - Tootmisjuht tahab näha päevapõhiselt töötajate poolt tehtud tundide hulka.
 - Tootmisjuht tahab töötaja peale klikkides näha töötajapõhist päevadest kõigi tehtud töödega
- Tootmisjuhile töötajapõhine kuuvaade
 - Tootmisjuht tahab näha kuupõhist tabelit, saamaks ülevaadet töötajast- töötajate ja päevade lõikes.
 - Tootmisjuht tahab esiletõstetuna näha liigraporteeritud tunde
 - Tootmisjuht tahab esiletõstetuna näha null-päevi
- Tellimuse töösse andmine
 - Tellimuse sisestamisel ei tohi tellimus automaatselt muutuda töökäsuks
 - Tootmisjuht kinnitab tellimuse töösse, kasutades selleks administreerimise veebiliidest

3.1 Kasutajalood

3.1.1 Funktsionaalsed nõuded

1. Kasutaja tahab sisse logida enda kasutajaga, et saaks enda tööd teha/jätkata
2. Kasutaja tahab näha töid eraldi kategooriate järgi, et saaks paremini navigeerida.
3. Kasutaja tahab näha põhilist infot töö kohta, et aru saada mis tööga on tegu.
4. Kasutaja tahab näha kui palju aega on tal juba teatud töö jaoks kulunud, et sellega arvestada.
5. Kui kasutaja räägib inglise keelt tahab kasutaja muuta keele inglise keelseks, et ta saaks paremini aru.
6. Kasutaja tahab kiiresti valmis tööde kategooria juurde saada, et aega säästa.
7. Kasutaja tahab otsida spetsiifilisi töid, selleks et aega säästa.
8. Kasutaja tahab valida mitu tööd, sest siis saab ta mitme asjaga korraga töötada.
9. Kasutaja tahab valida kõik tööd korraga, selleks et tegeleda kõikide töödega korraga ja säästa aega.
10. Kasutaja tahab näha kaua tal on hetkel tööde peale aega kulunud, selleks et sellega arvestada.
11. Kasutaja tahab vaadata tööplaani ilma et töid lõpetab.
12. Kasutaja töid lõpetades või pausile pannes tahab kulutatud aja ära jaotada töödele, mis on hetkel käima pandud.

3.1.2 Mittefunktsionaalsed nõuded

1. Rakenduse kasutamisel tuleb teha minimaalseid liigutusi, et töö oleks kiire
2. Rakendus peab olema kasutatav Android seadmetel
3. Rakendus peab olema kasutatav IOS seadmetel
4. Rakendus peab olema kasutatav tahvlitel
5. Rakendus ei tohi üleliia koormata mobiili
6. Rakendus ei tohi kasutada liigselt mobiili akut

4 Kasutatud tehnoloogiad

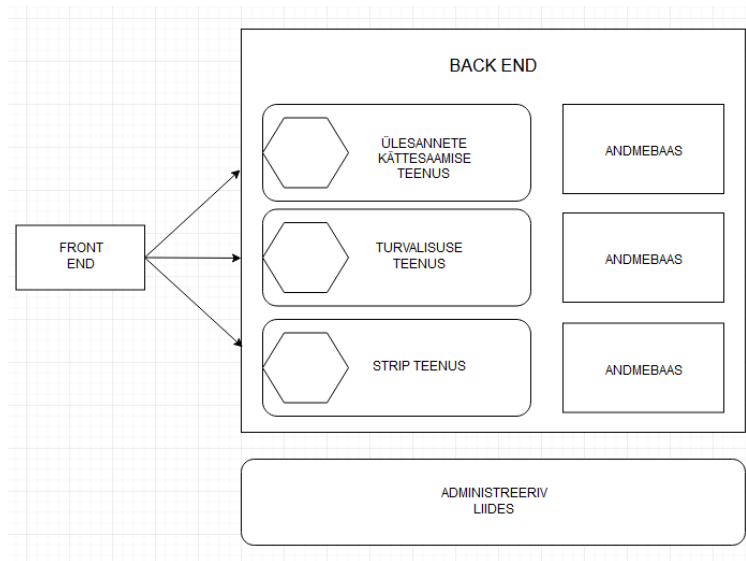
4.1 Vana rakendus

Algne versioon rakendusest valmis aastal 2015, ning sellel ajal oli rakenduse looja firmaks ProExpert. Vana rakenduse front-end oli tehtud Android NDKga ja back-end oli tehtud Play raamistikuga. Kuid siis juba selgus, et Ionic on palju kiirem ja mugavam kasutada, ning Android NDK vahetati Ionicu vastu välja. Play raamistik oli eesrindlik raamistik, ning sellega oli kerge teha REST teenuseid. ProExpert tegeles sellel ajal palju Java keelega, ning see kinnitas Play valiku.

4.2 Uus rakendus

Uus rakendus, mis valmis 2021, sai kirjutatud front-endi poole Angular ja Ionic raamistikkudel. Esimene lahendus oli kirjutatud ka Ionic raamistikul kuid see versioon oli väga aegunud ja sellepärast kirjutati kood ümber uuele versioonile. Eelmise versiooni back-end oli kirjutatud Play raamistikul, kuid see ei toetanud GraphQL, ning CQRS ja ES. LAGOM arendajaid oli vähe ja firma otsustas minna üle Nestjs peale, sest toetab GraphQL. Esialgsel tutvumisel oli NestJS väga hõlbust, ning selle kasutamisel ei esinenud probleeme. Rakenduses toimub ka kasutaja autentimine, ning see tehakse JWT abil.

Andmebaas disainiti DDD(Domeenipõhine disain)'ist lähtuvalt, ehk andmebaasi tabelid koondati tellimusteks, selleks, et vähendada relatsioone. Rakendus on mikroteenuste põhine.



Joonis 1. Lahenduse üldine arhitektuur

Angular on arendus platvorm, mis kasutab programmeerimiskeelena TypeScripti[13].

Platvormina Angular pakub komponentide põhise raamistikku, selleks et ehitada skaleeritavat rakendust.

Angularil on suur kollektsioon hästi integreeritud teek, mis katavad suure hulga funktsioone, sealhulgas ka marsruutimine, vormi haldus, kliendi-serveri suhtlus ja muud. Angular kuulutati välja ng-Europe konverentsil 22-23. Oktoobril 2014, kuid seda hakati muutma, ning lõpuks saadi versioon 2 mis sai avalikkusele kättesaadavaks alles 14. septembril 2016[14]. Peale seda on aastatega tulnud palju erinevaid versioone. Viimane versioon on Angular 13, mis ilmus neljandal novembril 2021. Rakenduse jaoks sai välja valitud Angular 12, sest oli kõige uuem versioon sel hetkel.

Ionic on täielik avatud lähtekoodiga SDK, mis on loodud Max Lynch, Ben Sperry ja Adam Bradley firmas Drifty Co 2013 ja on mõeldud hübriidsete mobiili rakenduste jaoks. Alguses

ehitati Ionic AngularJS ja Apache Cordova peale, kuid hiljem ehitati ümber selleks, et kasutaja saaks kasutada ükskõik mis kasutajaliidese disaini raamistikku.[15]

GraphQL on päringukeel API jaoks ja käitusaegne selleks et täita need päringud eksisteerivate andmetega. GraphQL pakub APIs olevate andmete täielikku ja arusaadavat kirjeldust, annab kliendile õiguse küsida täpselt seda mida nad vajavad ja mitte midagi enamat[16]. GraphQL on loodud Facebooki poolt ja esmane arendus algas 2012 ja esimene versioon tuli välja 2015 aastal. Facebooki idee oli luua uut tüüpi suhtlusviis andmebaasi ja kliendi vahel[17]. GraphQLi suurim eelis on see, et kasutab REST'i.

5 Valminud rakendus

Tänu Angulari modulaarsusele saab jagada rakenduse komponentideks ja teenusteks. Iga töö loetakse sisse, tööde kättesaamise teenusega. Teenuses lisatakse kõik andmed üheks paisktabeliks, mis viiakse ühest komponendist teiste, olenevalt kas seda on seal vaja. Projekt jaguneb kolmeks teenuseks ja kaheksaks komponendiks. Edaspidi on töö nimetatud ümber ülesandeks.

5.1 Teenused

5.1.1 Ülesannete kättesaamise teenus

Ülesannete kättesaamise teenus ehk *fetch-tasks.service.ts* on teenus mis küsib andmebaasilt kõikide ülesannete andmed, ning tagastab paisktabelina.

5.1.2 Turvalisuse teenus

Kasutaja sisse logimisega ja tema autentimisega tegeleb *sec-token.service.ts*. Turvalisuse teenuses kasutatakse JWT kasutaja autentimiseks.

5.1.3 Strip teenus

Ülesande hetkeseisu salvestamise ja tagastamise teenus ehk *strip-api.service.ts* vastutas selle eest, kui rakendus peaks kinni minema, siis andmed ei kaoks ega kustuks ja tööline saaks jätkata oma eelmiste ülesannetega sealt kus varasem töö pooleli jäi.

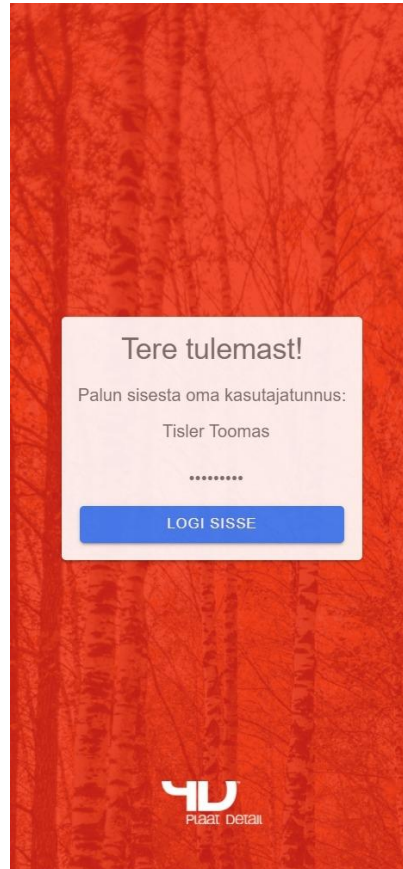
5.2 Front end komponendid

5.2.1 Rakenduse komponent

Rakenduse komponent on põhi komponent, mis ümbritseb tervet rakendust, siin seatakse ära mis keelt kasutab rakendus ja käivitab ülejäänud komponendid. Põhiliselt on rakenduse komponent raamiks tervele ülejäänud rakendusele.

5.2.2 Sisselogimise komponent

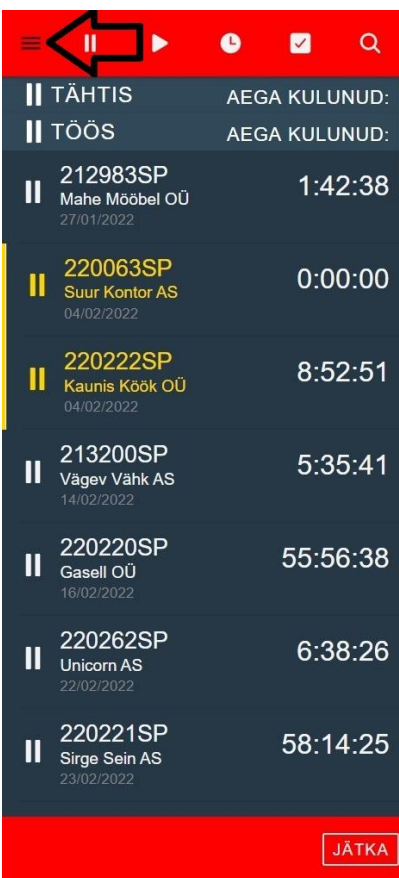
Sisselogimise komponent kuvatakse esimese asjana kui rakendus käivitatakse, sealt saab kasutaja sisse logida oma kasutajanime ja parooliga. Tänu Turvalisuse teenusele (mis tegeleb kasutaja autentimisega) lastakse kasutaja ülejäänud rakenduse juurde kui on tegu õige kasutajanime ja parooliga.



Joonis 2. Sisselogimise komponent

5.2.3 Menüü komponent

Menüü komponent asub ülesannete komponendis, ning on kõrvalt lahti tulev Menüü millega saab vahetada keelt ja välja logida, Välja logimine viib sisselogimise komponendini. Algselt on see ära peidetud ning vajutades Menüü nupule tuleb see külje pealt lahti.



Joonis 3. Menüü komponent

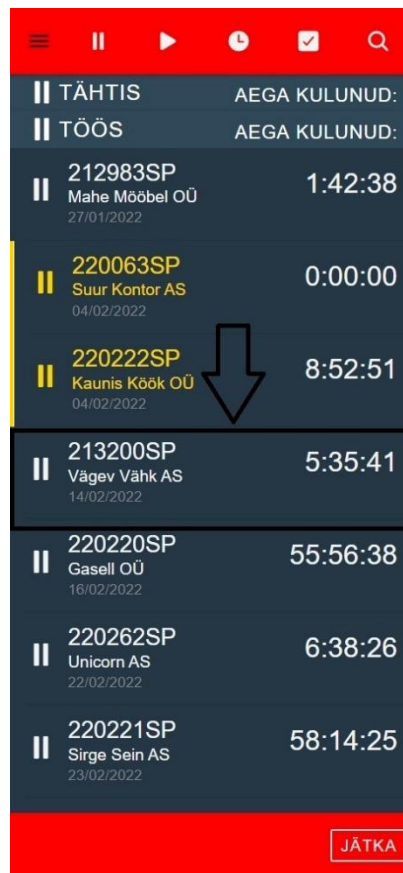
5.2.4 Ülesande komponent

Antud komponendis täpsustatakse ära milline näeb välja üks ülesanne. Antud komponenti kasutatakse üle rakenduse ja iga kord kui kasutatakse komponenti antakse ette muutujad.

Muutujateks on:

1. Töö pealkiri. See jaguneb kaheks töö number ja tehase nimi, hetkel on selleks SP, ehk Suur Paala
2. Kliendi nimi
3. Töö tähtaeg
4. Kui palju on juba varasemalt aega kulunud töö peale
5. Icoon, see teeb navigeerimise lihtsamaks.

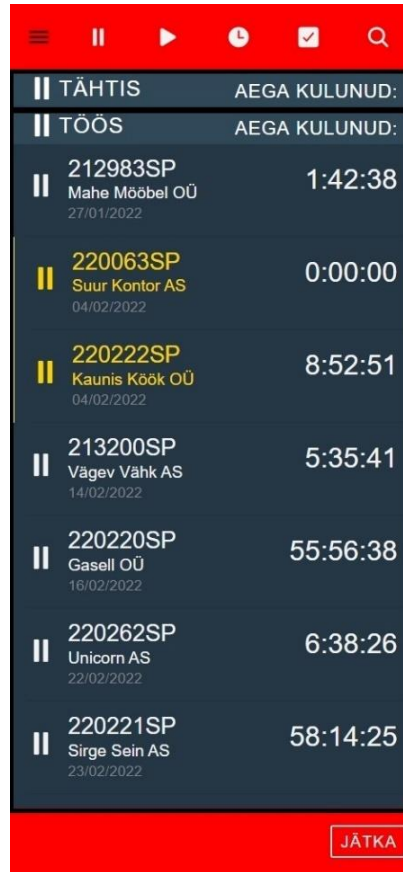
Olenevalt kus komponendis antud ülesannet kasutatakse on ülesandel vajutamise atribuut. See tähendab, kas ülesannet saab valida või ei saa



Joonis 4. Ülesande komponent

5.2.5 Ülesande nimekirja komponent

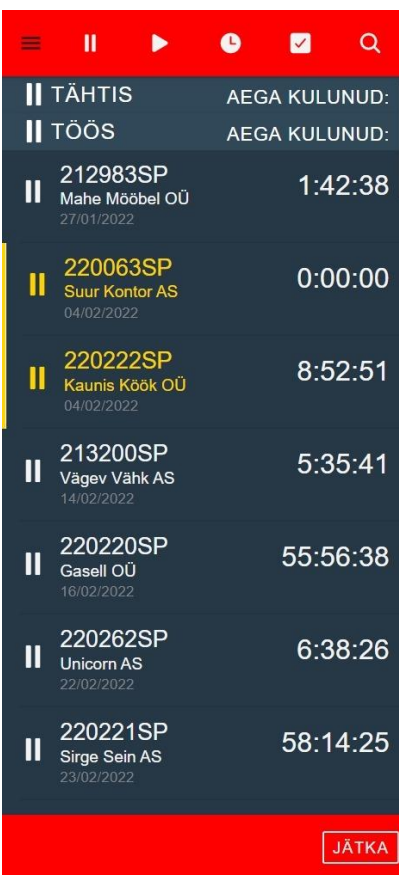
Ülesande nimekirja komponent ehitab ühe nimekirja, mis täidetakse ülesande komponentidega. Komponendile antakse ette ikooni nimi ja lahtri nimi. Kokku on neli ülesannete nimekirja, „Tähtis“, „Töös“, „Uued“ ja „Valmis“. Vajutades nimekirja päise peale, saad valida kõik antud nimekirjas olevad ülesanded.



Joonis 5. Ülesande nimekirja komponent

5.2.6 Ülesannete kuvamise komponent

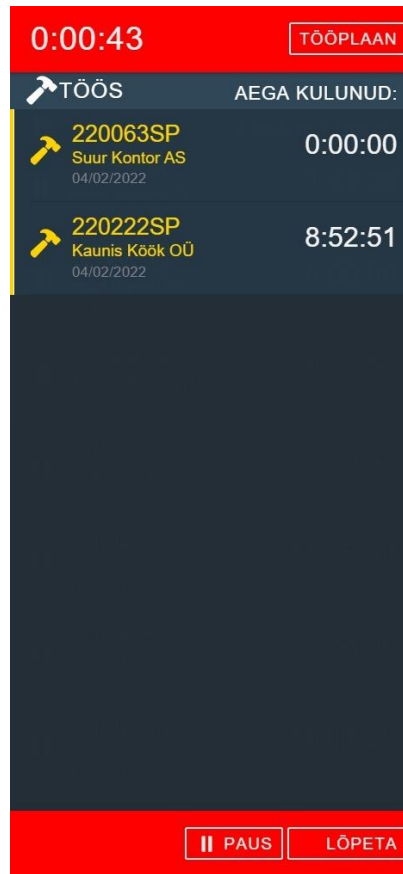
Antud komponent on kõige suurem komponent, komponendi päises on viis nappu, kõige vasakpoolsem on menüü napp, mis avab menüü komponendi. Siis tuleb neli navigeerimis nappu, mis liigutab vaate vastava nimekirja juurde. Viimane, ehk kõige parempoolsem napp avab otsingu lahtri, ning tööline saab otsida mingit spetsiifilist ülesannet. Edasi on sisu, sisus on neli nimekirja komponenti, mis on täidetud ülesande komponentidega. Ülesannete kuvamise komponendi jalutsis tekib napp kui on valitud vähemalt üks ülesanne, ning selle peale vajutades viiakse tööline hetke käimas olevate ülesannete komponenti.



Joonis 6. Ülesannete kuvamise komponent

5.2.7 Hetke käimas olevate ülesannete komponent

Kui jätkata/alustada ülesandeid siis viiakse kasutaja antud komponendini. Antud komponendi päises on näha aega, kui kaua on tööline hetkel tööd teinud, lisaks on nupp „TÖÖPLAAN“ see viib plaani komponendini. Antud komponendi sisus näeb kasutaja kõiki ülesandeid, mis kasutaja ennem oli valinud (ainult uued ülesanded, käimas olevad ülesanded ja tähtsad ülesanded) ja kui kaua oli varasemalt aega kulunud. Komponenti jaluses on nupp millega saab panna tööd pausile või lõpetada, ning see viib edasi ülesannete jaotamise komponendini.



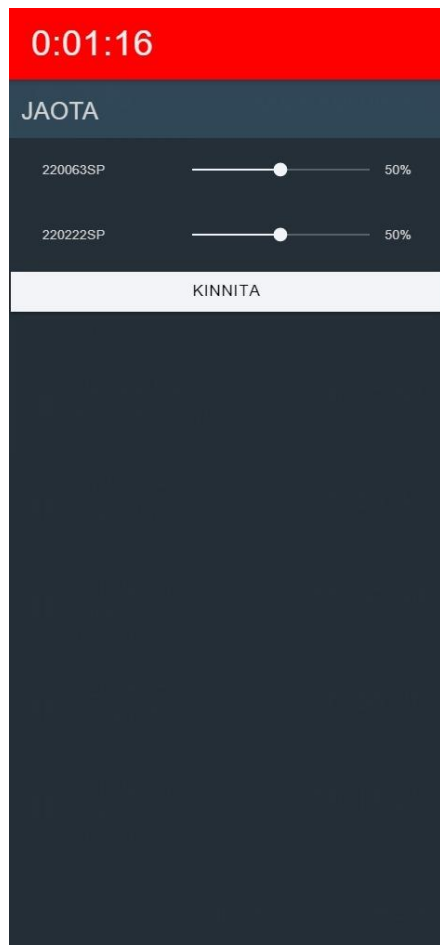
Joonis 7. Hetkel käimas olevate ülesannete komponent

5.2.8 Plaani komponent

Vajutades hetkel käimas olevate ülesannete komponendis nupule „TÖÖPLAAN“ viiakse teid plaani komponenti, seal on näha nimekirjad uutest, töös olevatest ja tähtsatest ülesannetest. Antud komponendis ei saa küll valida ühtegi ülesannet, kuid see annab ülevaate tööplaanist.

5.2.9 Ülesannete jaotamise komponent

Antud komponendi päises on hetkel ülesannete peale kulunud aeg. Antud komponendi sisus peab ülesannete mahu ära jaotama protsentuaalselt kõikide tööde vahel mida on lõpetatud või pausile pandud. Peale seda jagatakse kokku kulunud aeg protsentuaalselt.



Joonis 8. Ülesannete jaotamise komponent

6 Kokkuvõte

Bakalaureusetöö raames loodi front-end rakendus Plaat Detailile. Rakendusse sisenemisel on vaja kasutajal sisestada oma kasutajatunnus ja parool. Rakenduses saab tööline näha ülesandeid mis on kas pausil, valmis, lõpetatud või tähtsad. Tööline saab panna korraga mitu ülesannet käima ja kui ülesanded käivad saab tööline näha aega ja vaadata oma tööplaani. Kui ülesanded lõpetatakse jaotab töötaja ära kui palju ta mingi ülesande jaoks aega kulutas. Rakendusel on olemas ka back end ja inglise keelne tõlge. Tegemata on veel storybook lahendus, mis oli alguses olemas kuid eemaldasime.

Antud bakalaureusetöös on ülevaade kliendist ja mobiilirakenduse looja firmadest, ning kliendi esitatud nõuetest uuele mobiilirakendusele. Töös on tehtud ülevaade kasutatud tehnoloogiast algse 2016 aasta rakenduse kohta ja uue, aastal 2021 valminud rakenduse kohta. Lisaks annab töö ülevaate valminud rakendusest. Rakendus töötab nii nagu klient soovis, ning klient on väga rahul valminud rakendusega, seega bakalaureusetöö eesmärk on täidetud. Kahjuks Autori ja kliendi koostöö enam ei jätku.

Kasutatud kirjandus

- [1] <https://akit.cyber.ee/> (07.03.2022)
- [2] <https://et.build-repair.com/5808208-what-industry-4-0-means-for-people>(07.03.2022)
- [3] <https://www.sysdec.ee/saf.htm>(07.03.2022)
- [4] <https://et.portaldacalheta.pt/json-web-token-tutorial> (07.03.2022)
- [5] https://wikipedia.net/et/Domain-driven_design (08.03.2022)
- [6] <https://acronyms.thefreedictionary.com/NDK> (08.03.2022)
- [7] <https://developer.ibm.com/articles/an-introduction-to-command-query-responsibility-segregation/>
(08.03.2022)
- [8] <https://www.lagomframework.com/> (09.03.2022)
- [9] <https://storybook.js.org/> (09.03.2022)
- [10] <https://plaatdetail.ee/meist/#1608214091886-e3a83d91-735b> (12.10.2021)
- [11] <https://inspirators.eu/industry-4-0/> (09.01.2022)
- [12] Henno S. Homse maailma kirjanikud Tallinn: Spring Advertising, 2013
- [13] <https://angular.io/guide/what-is-angular> (10.02.2022)
- [14] [https://en.wikipedia.org/wiki/Angular_\(web_framework\)](https://en.wikipedia.org/wiki/Angular_(web_framework)) (07.03.2022)
- [15] [https://en.wikipedia.org/wiki/Ionic_\(mobile_app_framework\)](https://en.wikipedia.org/wiki/Ionic_(mobile_app_framework)) (09.03.2022)
- [16] <https://graphql.org/> (20.02.2022)
- [17] <https://et.wikipedia.org/wiki/GraphQL> (20.02.2022)

Lisad

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Andreas Poola,

(autori nimi)

- 1) annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose

Rakenduse “NobeTehas” arendamine,

(lõputöö pealkiri)

mille juhendaja on Vambola Leping,

(juhendaja nimi)

reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.

- 2) Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
- 3) Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
- 4) Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Andreas Poola

07.03.2022