

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Keit Arleen Kalle 163973 IABB

**ISETEENINDUSLIK PATSIENDI KOHALE
JÕUDMISE REGISTREERIMISE
RAKENDUS MEDITSIINIASUTUSELE:
ANALÜÜS JA PROTOTÜÜP**

Bakalaureusetöö

Juhendaja: Tarvo Treier
MSc

Tallinn 2019

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: Keit Arleen Kalle

20.04.2019

Annotatsioon

Antud bakalaureusetöö eesmärk on analüüsida iseteenindusliku patsiendi kohale jõudmise registreerimise rakendust ja luua vastav prototüüp. Analüüs põhineb autori tuttava perearstikeskuse nõudmiste põhjal, mida laiendatakse ka teistele Eesti perearstikeskustele.

Töö alguses kirjeldatakse protsessi olemust ja hetkeseisu nii meditsiinis kui ka teistes valdkondades. Otsitakse sarnasusi teiste valdkondadega ning leitakse parim lahendus, mis rahuldaks kohale registreerimise protsessi ka meditsiinivaldkonnas.

Töö tulemuseks on põhjalik analüüs ja veebirakenduse prototüüp, mis iga päeva hommikul saadab käesoleva päeva patsientidele emaili kohale registreerimiseks vajaliku lingiga, mis viib veebilehele, kus on kuvatud visiidi andmed. Kohale registreerimisel kuvab veebileht patsiendile vajaliku info ning protsess on edukalt lõppenud.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 25 leheküljel, 6 peatükki, 6 joonist.

Abstract

Patient self service check-in application for medical practices: analysis and prototype

The aim of this thesis is to analyze the need for a patient self service check-in application for medical practices in Estonia and to develop a prototype as a web application in order to provide a solution to medicine patient check-in process' needs. Analysis is based on the author's acquaintance's medical practice's needs, which can be expanded to other practices.

This thesis provides an introduction to different self service platforms by explaining their positive and negative features. The constructed analysis of the check-in process in medicine and other areas is the base of finding the needs of the process for the web application prototype.

Based on the introduced technologies and analysis of the check-in process, this thesis provides a realisation of a software application prototype. At the end of the thesis an analysis of results is formulated to evaluate the fulfillment of goals.

The thesis is in Estonian and contains 25 pages of text, 6 chapters, 6 figures.

Lühendite ja mõistete sõnastik

API	<i>Application Programming Interface</i> , programmiliides
HTMP	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i> , hüperteksti edastusprotokoll
JavaScript	Objekt orienteeritud programmeerimiskeel veebilehtede skriptimiseks
JSON	<i>JavaScript Object Notation</i> , lihtsustatud andmevorming
NoSQL	<i>Not Only SQL (Structured Query Language)</i> , andmebaasisüsteem, mis ei ole relatsiooniline
REST	<i>Representational State Transfer</i> , veebiteenuse arhitektuuri ülesehitamise viis
URL	<i>Uniform Resource Locato</i>
XML	<i>Extensible Markup Language</i> , dokumenditüüp

Sisukord

1 Sissejuhatus	9
1.1 Taust ja probleem	9
1.2 Ülesande püstitus.....	9
1.3 Metoodika.....	10
1.4 Ülevaade tööst.....	11
2 Iseteeninduse eesmärk ja lahendused	12
2.1 Iseteenindus	12
2.2 Iseteeninduse realiseerimise võimalused	12
2.2.1 Iseteeninduskiosk	13
2.2.2 Veebirakendus.....	13
2.2.3 Mobiilirakendus	14
3 Protsessi kirjeldus ja olemasolevad lahendused.....	15
3.1 Kohale registreerimine	15
3.1.1 Patsiendi kohale registreerimine Eesti meditsiinis.....	16
3.1.2 Kohale registreerimine meditsiinis välismaal	17
3.2 Kohale registreerimise lahendused teistes valdkondades.....	17
3.2.1 Lennundus	17
3.2.2 Spordiklubid.....	18
3.3 Kohale registreerimise protsessi nõuete analüüs.....	18
4 Analüüs.....	20
4.1 Sihtrühmad	20
4.2 Veebirakenduse analüüs	20
4.2.1 Funktsionaalsed nõuded	20
4.2.2 Mittefunktsionaalsed nõuded	21
4.2.3 Kohale registreerimise TO-BE protsessidiagramm.....	21
5 Realisatsioon	23
5.1 Tehnoloogiad.....	23
5.1.1 <i>Full stack</i> JavaScript arendus.....	23
5.1.2 ReactJS.....	24

5.1.3 NodeJS	24
5.1.4 ExpressJS	25
5.1.5 MongoDB.....	25
5.1.6 REST API.....	25
5.2 Rakenduse arhitektuur.....	26
5.3 API kirjeldus	27
5.3.1 Emaili vaade.....	28
5.3.2 Kohale registreerimise vaated	29
5.3.3 Visiidi toimimise info vaade	30
5.4 Hinnang rakendusele ja nõuete täitmisele	31
6 Kokkuvõte	33
Kasutatud kirjandus.....	34

Jooniste loetelu

Joonis 1. Kohale jõudmise registreerimise protsessi AS-IS diagramm	16
Joonis 2. Kohale registreerimise TO-BE diagramm	22
Joonis 3. Rakenduse arhitektuur.....	26
Joonis 4 Kehtiva visiidi kohale registreerimise vaade	Error! Bookmark not defined.
Joonis 5. Kehtiva broneeringu kohale registreerimise vaade	Error! Bookmark not defined.
Joonis 6. Kohale registreerimise vaade	30

1 Sissejuhatus

Viimastel aastatel on mitmes valdkonnas toimunud nii Eestis kui ka mujal maailmas äriprotsesside automatiseerimine. Erinevad teenused, mis varem nõudsid valdkonnateadmistega personali, kasutavad nüüdseks iseteenindusplatvorme, mistõttu teenusepakkujad saavad hakkama kasvavate mahtude juures väiksemate ressurssidega ning tarbijad säästavad aega, sest ei pea kusagil järjekorras seisma. Üldjuhul on iseteeninduslikud protsessid kiiremad ja efektiivsemad teenuse mitmekordsele kasutajale, kuna juba tuttava protsessi korral ei kulutata aega küsimuste vastamisele ja protsessi sammud on viidud miinimumini.

1.1 Taust ja probleem

Meditsiini valdkond muutub samuti üha enam digitaliseerituks kogu maailmas. Hetkel on Eestis kasutusel mitu erinevat digiregistratuuri sarnast teenust, kus on võimalik vaadata haiguslugusid ja broneerida soovitud visiidi aega erinevate tervishoiutöötajate juurde.

Antud töö autoril oli võimalus tutvuda ühe tervisekeskuse igapäevaste äriprotsessidega ning selle käigus hakkas silma aeg-ajalt tekkiv pikk ootejärjekord registratuuris. Selle põhjustas registratuuriõe suur ülesannete hulk, millest üks on patsientide kohale jõudmise registreerimine ning nende kabinetidesse suunamine. Kuigi patsient saab endale visiidi registreerida iseseisvalt digiregistratuuris, siis perearstikeskusesse kohale jõudes toimub protsess edasi manuaalselt.

1.2 Ülesande püstitus

Bakalaureusetöö põhieesmärk on lahendada patsiendi kohale registreerimise ajakulu Eesti meditsiinasutustes. Selleks analüüsitakse visiitidele kohale jõudmise registreerimise protsessi ja hetkeseisu nii Eestis kui ka välismaal. Kogutud info põhjal tehakse järeldused ja realiseeritakse iseteeninduslik patsiendi kohale registreerimise rakenduse prototüüp.

Töö põhieesmärgi täitmiseks püstitatud alameesmärgid on sõnastatud järgmiselt:

- Anda ülevaade iseteenindussüsteemide kasulikkusest ja levinumatest lahendusviisidest
- Anda ülevaade ning analüüsida kohale jõudmise registreerimise protsessi ja selle olulisust meditsiinivaldkonnas
- Analüüsida teistes valdkondades sarnasele protsessile loodud lahendusi ning hinnata kas need oleksid sobilikud ja vajalikud ka meditsiinasutuses
- Viia läbi veebirakenduse analüüs, arvestades kohale jõudmise registreerimise protsessi infovajadusi ja töötavaid lahendusi muudes valdkondades
- Realiseerida ja testida kohale registreerimist võimaldava rakenduse prototüüp

1.3 Metoodika

Bakalaureusetöö valmib järgmistes etappides:

- Iseteeninduslike süsteemide üldiste põhimõtete uurimine
- Patsiendi kohale jõudmise registreerimise protsessi analüüsimine Eesti meditsiinis käesoleval hetkel ning rakenduse vajaduste sõnastamine
- Valmislahenduste alusel kohale registreerimise protsessi rahuldava tarkvaralahenduse sõnastamine
- Analüüsi põhjal kirjapandud nõuetele vastava rakenduse loomine kasutades uuritud tehnoloogiad
- Loodud rakenduse funktsionaalsuse testimine ning soovitud tulemuse saavutamise hindamine
- Lõpptulemuste formuleerimine

1.4 Ülevaade tööst

Iseteenindusest ja selle realiseerimise võimalikest lahendustest annab ülevaate teine peatükk. Töö kolmas peatükk seletab lahti kohale registreerimise protsessi ja selle vajalikkuse ning selgitab välja loodava rakenduse infovajadused, tuginedes avalikult kätte saadavale infole ning autori isiklikule kokkupuutele kohale registreerimise protsessiga Eesti meditsiinisüsteemis. Lisaks analüüsitakse kolmandas peatükis iseteeninduslikke lahendusi kohale registreerimisest teistes valdkondades ning saadud info põhjal sõnastatakse veebirakenduse funktsionaalsed nõuded. Neljandas peatükis tehakse meditsiinasutuses kohale jõudmise registreerimise vajaduste põhjal antud töö käigus valmiva veebirakenduse analüüs. Viies peatükk kajastab veebirakenduse realiseerimiseks kasutatud tehnoloogiate valikut, valminud rakenduse arhitektuuri ja ülesehitust. Samuti analüüsitakse tulemusi ning hinnatakse soovitud eesmärkide täitmist.

2 Iseteeninduse eesmärk ja lahendused

Paljude äriprotsesside automatiseerimiseks kasutatakse erinevaid iseteeninduslikke lahendusi. Järgnevas peatükis analüüsitakse iseteeninduse eeliseid ja puudusi ning võimalikke viise, kuidas erinevaid protsesse automatiseerida.

2.1 Iseteenindus

Üha enam väheneb klientide otsene suhtlus teenusepakkujatega tänu iseteenindusplatvormidele. Tänapäevase kiire elutempoga maailmas muutub suur hulk teenuseid üha enam tehnoloogiapõhiseks, sest vajalikku tööjõudu on liiga vähe, kes rahuldaks kasvavat kliendibaasi. Suurem hulk tarbijaid kasutabki ainult tehnoloogiat, et jõuda soovitud teenuse lõpptulemuseni.

Iseteenindusplatvormid on tehnoloogilised liidesed, mis võimaldavad kliendil tarbida soovitud teenust ilma toetava personali abita. Esimese suurema uuendusena võeti kasutusele panganduses sularahaautomaadid, järgmiseks ilmusid hotellidesse lahkumise registreerimist võimaldavad kioskid ja nendele järgnesid ülejäänud internetis toimivad teenused. Iseteenindusvõimalust loetakse üheks pikaajalise edu eelduseks, tuues turule uut toodet või teenust. [1]

Iseteeninduse kasutamist hakati rohkem uurima 1980. aastatel. Algselt kasutasid seda rohkem nooremad, kõrgema haridustasemega ja madalamapalgalised, kuid hilisemalt on selgunud, et ka vanemad generatsioonid hindavad tihti iseteeninduse võimaluse olemasolu. Iseteeninduse peamine eelis kliendile on vähenenud ajakulu soovitud tulemuse saavutamiseks. Teenusepakkujad saavad hakkama väiksemate tööjõukuludega, samal ajal kui klientide arv kasvab. Iseteeninduslikud lahendused hakkavad toimima enamasti siis kui loodud platvorm on kasulik nii teenusepakkujale kui kliendile. [1]

2.2 Iseteeninduse realiseerimise võimalused

Iseteenindussüsteemide loomiseks on erinevaid viise, mis kõik on tehnoloogiapõhised, selleks, et vähendada ettevõtte tööjõukulutusi või rakendada antud tööjõudu muudes

ülesannetes ning kliendile annab see võimaluse läbida protsessi vähema ajakuluga. Järgnevates alapeatükkides analüüsitakse erinevaid võimalusi, kuidas rakendada iseteenindust ning mis on nende eelised ja puudused. Valik tehti iseteenindus võimaluste populaarsuse alusel, et töö käigus loodav rakendus oleks kindel ja mitte liialt keeruliste nõudmistega.

2.2.1 Iseteeninduskiosk

Iseteeninduskiosk on igasugune füüsiliselt eksisteeriv eesmärgipõhiselt paigaldatud seade ühekordseks kasutuseks. Esimeste näidetena loodi pangaautomaadid ja toidupoodide kassasüsteemid, kus klient saab ilma personali abita vajaliku tegevuse sooritada. Tänapäeval on võimalusi kordades rohkem, selleks on vajalik puutetundliku ekraaniga seade ja vastav tarkvara, mis pakub teenust.

Paigaldatud kioski üks eelistest on see, et kliendil pole vaja isiklikku seadet kasutada ega ühtegi tarkvara alla laadida. Tarkvara, mis on loodud kioskis kasutamiseks saab olla loodud ainult ühele kindlale operatsioonisüsteemile, mis lihtsustab arendusprotsessi. Samas on võimalik paigaldada kioskisse veebibrauseripõhine tarkvara, kuigi antud lahenduse korral võib ilmned raskusi isikutuvastamise protsessiga.

Kioski paigaldamine ja hooldamine nõuab vastavat kapitali, mis on üheks suuremaks puuduseks, kuna riistvara vananeb suhteliselt kiiresti, siis teatud aja tagant tuleb näiteks paigaldada uued kioskid. Samuti võib iseteeninduskioski juurde tekkida järjekord, juhul kui kasutaja ei ole süsteemiga tuttav ja vajab abi, seetõttu peavad teised kliendid ootama ja iseteenindus on kaotanud ühe oma suurtest eelistest, ajakulu vähendamise.

2.2.2 Veebirakendus

Veebirakendus on mistahes rakendus, mis kasutab serveritega ühendamiseks HTTP protokolle. Paljud teenused asuvad veebirakendustes, sest need on igale internetikasutajale lihtsasti kättesaadavad. Selliste rakenduste arendamine on tehtud väga lihtsalt, sest on palju erinevaid raamistikke ja tööriistu, mida saavad vabalt kasutada kõik huvilised.

Veebirakendused on operatsioonisüsteemist ja arvutitüübist sõltumatud, mis on nende suur eelis. See tähendab, et veebirakenduse kasutamiseks ei ole vaja seda alla laadida, kuna igas arvutis on eelduste kohaselt olemas veebibrauser. Samuti on nendele lihtne

lisada täiendavaid funktsioone ja teha uuendusi, sest lõppkasutajad näevad muudatusi peatselt peale veebiserveri uuendamist.

Veebirakenduse loomisel on veidi keerukam isikutuvastamise protsessi loomine. Selleks on erinevaid võimalusi, saab luua rakendusele kasutaja sisselogimise funktsiooni või tuvastada klienti ID-kaardi tarkvara abil. Eksisteerib lahendus, kus veebibrauser tuvastab kliendi sessiooni ja ainult nime või emaili alusel, seega isikutuvastamise lahenduse loomine võib olla üsna lihtne.

2.2.3 Mobiilirakendus

Mobiilirakendus on tarkvara, mis on arendatud kasutamiseks nutitelefonides ja tahvelarvutites. Tihti need kasutavad veebiserverites olevaid teenuseid, et teha kasutajale suuremahuliste veebirakenduste kasutamine mugavamaks mobiilis.

Mobiilirakendusi saab arendada erinevatel viisidel: funktsionaalne- ja hübriidrakendus. Funktsionaalsed mobiilirakendused on kindlast operatsioonisüsteemist sõltuvad, mis tähendab, et enne arenduse alustamist tuleb valida, kas antud rakendus ehitatakse Android või iOS operatsioonisüsteemi peale. Hübriid mobiilirakendus on vahepealne arendusvõimalus veebirakendusest ja funktsionaalsest mobiilirakendusest. See võimaldab ühise koodibaasiga katta suurema osa mobiilsete operatsioonisüsteemide turust ja samal ajal pakub võimalust kasutada tarkvarast sõltuvaid komponente nagu kaamera ja graafika API.

Selleks et kasutada mobiilirakendust tuleb antud tarkvara alla laadida oma seadmesse, mis kulutab mahtu ja kaasab endaga mitmeid turvariske. Samuti nõuab mobiilirakenduse arendamine platvormipõhiseid programmeerimise teadmisi, mistõttu on ka arendajate leidmine veidi keerukam.

Isikutuvastamise funktsioon antud lahenduse korral saaks olla kas installatsioonipõhine või sarnaselt veebirakendusele tuvastatakse kasutaja eelnevalt loodud kasutajaga sisse logimine. Mõlemal lahendusel on eeliseid ja puuduseid, installatsioonipõhine tuvastus võimaldaks ühest seadmest vaid ühe kasutajaga rakenduse kasutamist, aga samas on mugav lahendus, sest kasutaja ei pea pidevalt sisse logimise tunnuseid sisestama. Igal korral rakendust kasutades sisse logimine on kasutajale tülikas, aga kohati turvalisem variant.

3 Protsessi kirjeldus ja olemasolevad lahendused

Käesolev peatükk analüüsib kohale registreerimise protsessi ja sellest tulenevalt tehakse järeldused, millised on vajalikud nõudmised bakalaureusetöö raames loodud rakenduseks. Viiakse läbi kohale registreerimise protsessi infovajaduste analüüs autori isikliku kogemuse ja artiklites leiduva info alusel.

Hinnatakse kohale registreerimise hetkeolukorda erinevates valdkondades teenusepakkujate veebilehtedelt saadavate andmete põhjal. Teiste valdkondade analüüsis tuuakse välja olemasolevate lahenduste eelised- ja puudused.

Protsessi kirjelduse ja olemasolevate süsteemide analüüsi peatükis selgunud loodava teenuse vajaduste põhjal viiakse läbi veebirakenduse analüüs, mida aluseks võttes luuakse bakalaureusetöö raames valmiv veebirakendus.

3.1 Kohale registreerimine

Kohale registreerimine on protsess, kus klient informeerib teenusepakkujat oma saabumisest. Teatud teenuste osutamisel annab antud info teenusepakkujale võimaluse teha ära vajalikud ettevalmistused, et kliendi jõudes sujuks protsess kiiremini. Juhul kui klient ei ole teinud kohale registreerimist piisavalt varakult, saab teenuseosutaja pakkuda vabanenud aega kellelegi teisele.

Eduka kohale registreerimise eelduseks on vastava broneeringu olemasolu. Teiseks oluliseks tingimuseks on kohale registreerimise sooritamine määratud ajavahemikus, mis sõltub pakutavast teenusest. Kui need kaks tingimust on täidetud, siis on mitmeid erinevaid võimalusi, kuidas saab klient kohale registreerida.

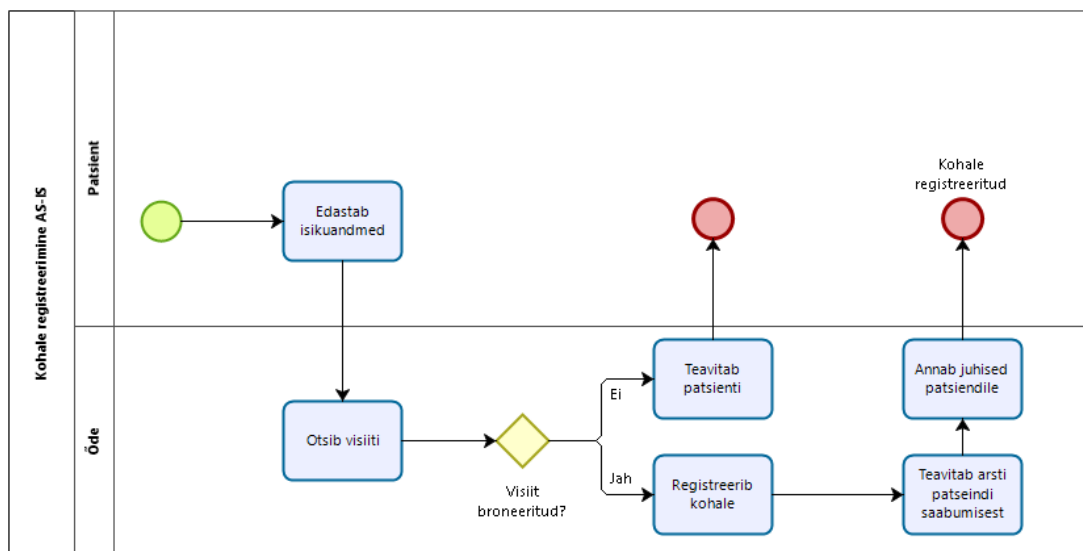
Antud protsessi on võimalik automatiseerida enamikes valdkondades. Selleks on vaja ligipääsu kliendi ja broneeritud teenuse andmetele ning käesolevas bakalaureusetöös analüüsitakse, kuidas toimiks kohale jõudmise registreerimise automatiseerimine meditsiinivaldkonnas.

3.1.1 Patsiendi kohale registreerimine Eesti meditsiinis

Eestis on meditsiinasutustes patsientide kohale registreerimise protsess manuaalne ja seetõttu eeldab registratuuripersonali aega ja kohalolu. Kohale registreerimise protsess algab patsiendil meditsiinasutusse sisenemisel, sest iga teenuse jaoks on vaja personali teavitada, et klient on kohal.

Patsient läheb registratuuri ja esitab vajalikud andmed, suuremates asutustes on isikut tõendava dokumendi esitamine kohustuslik, aga väiksemates keskustes piisab ka patsiendi nimest. Kohale registreerimine ilma isikut tõendava dokumendi või ühegi teise isikutuvastamise meetodita on patsientidele mugav, aga samas tekib turvalisuse kadumise risk. Õde otsib kehtivat visiiti süsteemist ning visiidi puudumise korral teavitab sellest patsienti ja kohale registreerimise protsess on lõppenud ebaõnnestunult.

Juhul kui visiit on broneeritud, siis registreerib õde süsteemis, et patsient on saabunud. Süsteem saadab arstile vastava teavituse. Õde teavitab patsienti, et visiit on registreeritud ja toimub kabinetis number X ning edastab muu vajaliku info. Sellega on kohale registreerimise protsess lõppenud.



Powered by
bizagi
Modeler

Joonis 1. Kohale jõudmise registreerimise protsessi AS-IS diagramm

3.1.2 Kohale registreerimine meditsiinis välismaal

Teatud riikides on ka meditsiinis patsiendi kohale jõudmise registreerimine mingi tasemeni automatiseeritud. Tervisekeskustesse ja haiglatesse on paigaldatud iseteeninduskioskid, mis võimaldavad patsiendil kohale registreerida.

Eeldusel, et patsiendil on visiit varasemalt broneeritud, kas veebikeskkonnas või registratuuris, suunatakse meditsiiniastutusse sisenedes iseteeninduskioski juurde. Süsteemid on enamasti inglisekeelsed ja patsiendil palutakse sisestada sugu ja sünnikuupäev, seejärel tagastatakse isikuandmed, broneeritud aeg ja ruum, kus visiit toimub. Sarnase iseteenindussüsteemi puuduseks võib lugeda turvalisust, kuna protsessi käigus ei küsita isikut tõendavat dokumenti. See tähendab, et suvaline isik võib sisestada kellegi teise nime ning seetõttu saada ka ebasobiva ravi. [2]

3.2 Kohale registreerimise lahendused teistes valdkondades

Analüüsides hetkel kasutatavaid lahendusi kohale registreerimisest, selgub, kas antud töös pakutav lahendus suudaks konkureerida juba turul olemasolevatega. Tuleb välja selgitada, millised on olemasolevad võimalused ja nende põhjal luua selged hinnangud loodavale rakendusele.

3.2.1 Lennundus

21. sajandi alguses muutus maailm ja inimeste elu kiiremaks ja seega ka mobiilsemaks. Üha enam on vaja liikuda riikide vahel nii äri kui ka personaalsetel eesmärkidel. Lendamise nõudlus kasvas hulgaliselt ja lennujaamadesse tekkisid pikad järjekorrad, mistõttu võeti kasutusele erinevate teenuste jaoks iseteenindusplatvormid.

Kõige edukam lahendus oli kohale registreerimise iseteeninduskioskite kasutusse võtmine, et vähendada lennule registreerimise järjekordi, sest iseteeninduse üks eelistest on kiirem protsessi toimumise aeg. Mõni aeg hiljem toodi turule ka veebirakendused, mis võimaldavad lennule kohale registreerimist teha isiklikust seadmest. Lennule registreerimiseks tuleb end tuvastada, kas isikut tõendava dokumendiga või lennupileti broneeringu numbriga, mille alusel suunatakse klient edasi kohale registreerimiseni.

Iseteenindusplatvorm annab kliendile hulganisti lisaväärtusi, pakub rohkem kontrolli teenuse üle ning seetõttu muudab privaatemaks. Iseteeninduses saab klient samuti

broneerida pileteid, teha kohale registreerimist lennule ja hiljem soovi korral printida piletid lennujaamas asuvast kioskist. Juhul kui lennuag muutub või on tühistatud, siis teavitab rakendus ka klienti, mis on lisaväärtus, mida varasemalt polnud võimalik lahendada. Mainitud võimalused säästavad kliendile hulganisti aega, eeldusel, et süsteemid toimivad kiirelt ja efektiivselt. Samuti pakub iseteenindusplatvorm erinevaid suuri eeliseid lennufirmadele. Teenus muudetakse automatiseerituks, mis vähendab tööjõukulusid või annab võimaluse ressursse arukalt ümber paigutada. [3]

3.2.2 Spordiklubid

Spordiklubid kasutavad samuti erinevaid iseteenindusviise selleks, et veenduda, kas broneeringu teinud klient on kohale jõudnud, vastasel juhul loovutatakse koht trennisaalis.

Kohale jõudes tuvastab klient end vastava spordiklubi kaardiga iseteenindussüsteemis ja ekraan kuvab eelnevalt tehtud broneeringu. Klient saab selle kinnitada või tühistada. Esimesel juhul trükitakse pilet, mis tuleb esitada treenerile ja teisel juhul vabastatakse koht süsteemis. Kui klient ei ole saabunud teatud kindlaks määratud aeg enne trenni algust, siis vabastatakse koht.

3.3 Kohale registreerimise protsessi nõuete analüüs

Analüüsid ja hinnates teistes valdkondades kasutusel olevaid lahendusi saab tuua paralleele ka meditsiinivaldkonda ning teha järeldusi nõuetest, mida kasutada antud bakalaureusetöö raames valmivas rakenduses.

Kõik eelpoolmainitud iseteenindusplatvormid tegelevad mingil tasemel süsteemi kasutava isiku tuvastamisega. Arvestades, et pettuse risk tervisekeskusesse kohale registreerimise protsessi juures on väike, sest patsient ise ei saaks sellest kasu. Siiski kuna võimalus selleks eksisteerib, siis mingil tasemel isikutuvastamise funktsioon peab olema ka loodaval rakendusel.

Paljud lennufirmad saadavad üks kuni kolm päeva varem teavituse, kas meili, sõnumi või rakenduse teel vastava lennupileti omanikule, et kohale registreerimine on nüüd avatud. Enamasti on seal ka juhised või link protsessi läbimiseks. Ka meditsiinivaldkonnas võiks iseteenindusplatvorm toimida sarnaselt, aga selleks, et arst oleks teavitatud kui patsient

on füüsiliselt keskusesse kohale jõudnud, siis peaks kohale registreerimine toimima vaid paarkümmend minutit kuni üks tund enne visiidi algust. Sarnane olukord, kus patsient kinnitab, et on keskuses kohal juba paar tundi enne visiidi aega, on vähe tõenäoline tekkima, sest ka patsient ise ei saa sellest suurt kasu.

Töö käigus valmiva tarkvara loomisel tuleb arvesse võtta, et osa meditsiiniinstituuti laseb broneeringut teha samale päevale ja ka antud patsientide kohale jõudmist on vaja registreerida. Seega peab rakendus võimaldama meeldetuletuse saatmist mitu korda päevas.

Suur eelis iseteenindustarkvaral on võimalus panna süsteem tööle mitmes keeles, mis tähendab, et rakendust saavad kasutada ka rahvusvahelised kliendid. Tervishoiutöötajad ei oska tihti rahvusvahelisi keeli ja seega muutub manuaalne kohale registreerimine keeruliseks. Probleemi lahendaks hästi loodud tarkvara, mis võimaldaks kasutajal määrata soovitud keele ning oleks võimeline edastama sõnumeid vastaval viisil.

Iseteenindustarkvara üks eesmärkidest on teenuse tarbimise lihtsustamine kliendile, seega peab olema üheselt mõistetav ja kerge kasutada, et ei valmistaks raskusi isegi vanemale generatsioonile.

Kirjeldatud nõudmistest järeldades ja samal ajal arvestades patsiendi mugavusega, oleks kõige mõistlikum luua veebirakendus, mis võimaldaks kohale jõudmise registreerimist teha isiklikust seadmest. Meditsiiniinstituutidele oleks see tõenäoliselt samuti kasulikuim variant, sest ei ole vaja paigaldada seadmeid, mis vajaksid korralisi hooldusi, mis omakorda nõuaks osa keskuse eelarvest.

4 Analüüs

Käesolevas peatükis analüüsitakse, millistele nõudmistele peab vastama loodav rakendus. Nõuete analüüsi aluseks võetakse info kogumise ja teiste valdkondade lahenduste analüüsi käigus ilmnunud kohale registreerimise süsteemi vajadused.

4.1 Sihtrühmad

Käesoleva bakalaureusetöö fookus on luua iseteeninduslik rakendus, mis rahuldab meditsiini-asutustesse kohale registreerimise protsessi. Seetõttu on antud töö raames loodaval teenusel järgmised sihtgrupid:

- Tervise- ja perearstikeskused, kes kasutavad veebiteenuse funktsionaalsust kohale registreerimise mugavdamiseks
- Tarkvaralahenduste pakkujad, kes kasutavad teenuse funktsionaalsust tervisekeskuste veebikeskkondade arendamisel

Lisaks eelpool mainitud peamistele sihtgruppidele on antud bakalaureusetöö tulemusel rakenduseedasiarendamise võimalused, mis omakorda laiendab kasutajaskonda. Antud platvorm võiks laieneda kogu Eesti meditsiini raames ja liituda 2019. aastal avatud uue digiregistratuuri süsteemiga. Selliselt on võimalik luua lisafunktsionaalsusi vastavalt meditsiini-valdkonnale ning potentsiaalselt suurendada teenust haldava ettevõtte kasumit.

4.2 Veebirakenduse analüüs

Rakenduse analüüs esitab töös loodava teenuse funktsionaalseid ja mittefunktsionaalseid nõudeid ning kirjeldab patsiendi kohale jõudmise registreerimise protsessi uut lahendust.

4.2.1 Funktsionaalsed nõuded

Rakenduse funktsionaalsed nõuded on järgmised:

- Rakendus peab võimaldama pärida patsiendi ja visiidi andmeid

- Rakendus peab võimaldama muuta visiidi staatust
- Rakendus peab võimaldama isikutuvastamist ehk garanteerima õige visiidi andmetega emaili saatmist visiidil osalevale isikule
- Rakendus peab võimaldama kliendi asukoha tuvastamist, et kohale jõudmise registreerimine saaks toimuda ainult meditsiinasutuse läheduses
- Rakendus peab võimaldama meeldetuletuse saatmist, mis sisaldab unikaalset kohale jõudmise registreerimise linki, hiljemalt üks tund enne visiidi algust
- Rakendus peab visiidi staatuse muutmisel saatma vastava teate arstile
- Rakendus peab vastavalt visiidi staatusele kuvama patsiendile korrektse info: visiidi toimumise asukohta, kabineti numbri ja arsti nime
- Rakendus peab vigade korral teavitama rakendust võimalikult täpselt ja informatiivselt

4.2.2 Mittefunktsionaalsed nõuded

Rakenduse mittefunktsionaalsed nõuded on järgmised

- Rakenduse lähtekood peab olema võimalikult lihtne ja sellega looma eelduse lisafunktsionaalsuste realiseerimiseks
- Rakendus peab realiseerima minimaalsel tasandil turvalisuse

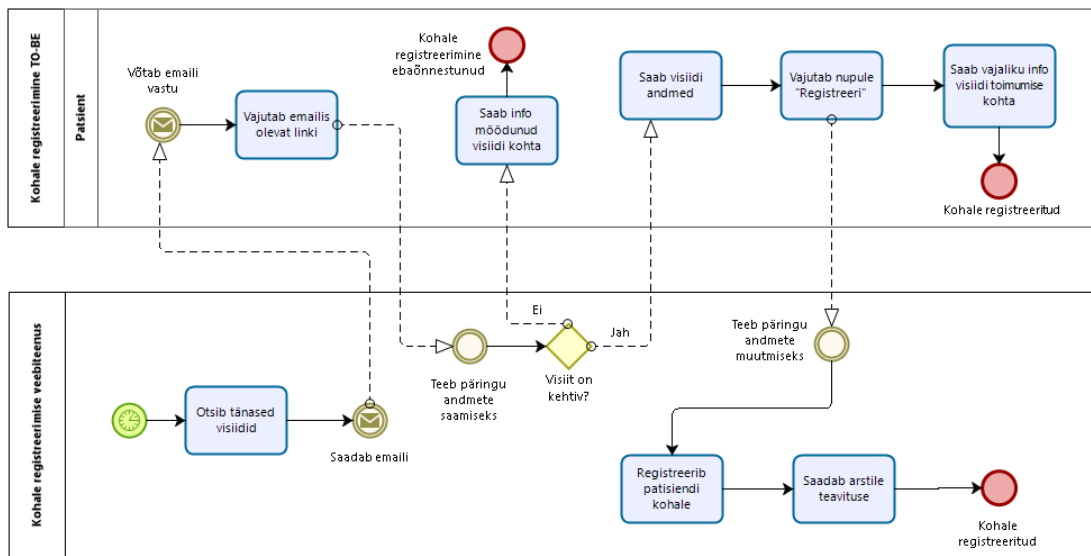
4.2.3 Kohale registreerimise TO-BE protsessidiagramm

Uue süsteemi kohaselt hakkab kohale registreerimise protsess loodud teenuse poole pealt, kui server saadab emaili koos vastava infoga õigele patsiendile. Email sisaldab unikaalse tunnusega linki, mis võimaldab patsiendil end antud visiidile kohale registreerida.

Patsiendi jaoks algab registreerimine, kui ta saab kätte serveri poolt saadetud emaili. Klippides emailis olevale lingile, kontrollib rakendus visiidi kuupäeva ja kellaaega ning kui visiidi aeg on minevikus, siis kuvatakse patsiendile vastav teavitus HTML lehena ja kohale registreerimine ei ole võimalik ning protsess on lõppenud ebaõnnestunult. Juhul kui visiit on kehtiv avatakse veebibrauseris lehekülg, kus on kuvatud visiidi peamine info

ja nupp „Registreeri“. Eeldusel, et info on korrektne, vajutab klient nupule ja talle kuvatakse kohale registreerimise kinnitus, koos muu vajaliku infoga, mida varasemalt edastas patsiendile registratuuris protsessi läbi viiv õde.

Nupule vajutades saadab veebileht API kaudu info serverile tagasi ja andmebaasis muudetakse vastavat kirjet, et patsient on kohale registreeritud. Samaaegselt saadab rakendus arstile teavituse, et patsient on saabunud. Kohale registreerimise protsess on edukalt lõppenud.



Powered by
bizagi
Modeler

Joonis 2. Kohale registreerimise TO-BE diagramm

5 Realisatsioon

Käesoleva töö realiseerimise peatükk kirjeldab veebirakenduse arendamist eelmises peatükis sooritatud analüüsi põhjal.

Töö tulemusena realiseeritakse analüüsi peatükis kirjeldatud nõuete kohane veebirakenduse prototüüp, mis on meditsiinasutuses patsiendi kohale jõudmise registreerimise funktsionaalne süsteem. Rakenduse realiseerimisel on loodud kõik eeldused, et hiljem saaks arendada prototüüpide põhjal terviklik lahendus.

Registreerimistarkvara lahenduses luuakse REST protokolliga järgiv teenus, mis kasutab lokaalset MongoDB andmebaasi visiitide info saamiseks ning loodud API on ehitatud NodeJS programmeerimiskeeles. Veebirakenduse kliendirakendus on kirjutatud kasutades ReactJS raamistikku.

5.1 Tehnoloogiad

Uue rakenduse prototüübi loomisel kasutatud tehnoloogiate valikul lähtuti ennekõike sellest, et tegemist oleks populaarsete ja samas usaldusväärsete tehnoloogiatega. Kasutades juba pikema ajalooga tehnoloogiaid, on küsimustele vastuste leidmine lihtsam. Lisaks muutub rakenduse edasiarendamise kergemaks, kuna tuntud tehnoloogiate kasutamine lihtsustab leida arendajaid, kes juba oskavad neid kasutada

5.1.1 *Full stack* JavaScript arendus

JavaScript on uuringutekohaselt juba pikaajaliselt olnud liidripositsioonil populaarsemate programmeerimiskeelte seas. Suured korporatsioonid nagu PayPal, Netflix ja Twitter kasutavad antud meetodit osa oma toodete arendamiseks, millest saab järeldada, et tegemist on usaldusväärse arendusviisiga. [4]

Full stack JavaScript programmeerimine aitab kaasa arendustiimi ühtsusele, sest kui kogu veebirakendus on kirjutatud JavaScriptis, siis nii *frontend* kui *backend* arendajad mõistavad kogu rakenduse koodi. Uuringutest on selgunud, et kui kogu kood kasutab

samu mudeleid ja teeke siis on võimalik vähendada koodiridade arvu 33% võrra. JavaScripti raamistikud on enamasti tasuta ja avalikud vabaks kasutuseks kõigile, seega ei teki projektis ka lisakulutusi kallitele arendusvahenditele. JavaScripti kasutamine teeb JSON-tüüpi andmevahetuse kordades kiiremaks, kuna rakendus kulutab vähem aega andmetüüpide teisendamisele. [5]

5.1.2 ReactJS

ReactJS on Facebook'i poolt arendatud *frontend* arenduse teek, mis on avatud lähtekoodiga, seega selle kasutamiseks ei ole ühtegi piirangut. Antud raamistiku eesmärk on lihtsustada interaktiivsete olekutega ja uuesti kasutatavate kasutajaliidese komponentide loomist. [6]

ReactJS-i peamine eelis teiste sarnaste raamistike ees on veebilehel andmete ja komponentide muutmine ilma terve lehe värskendamiseta. Projektis, kus kasutatakse ReactJS-i arendatakse JSX-iga, mis on lihtne objekt orienteeritud programmeerimiskeel disainitud veebibrauseritele. See on kiirem kui tavaline JavaScripti kood ning turvaline, sest kood on staatiliselt kirjutatu. [7]

5.1.3 NodeJS

Antud töö raames loodud rakenduse *backend* loogika arendamiseks valiti NodeJS. Tehnoloogiat tutvustati esmakordselt 2009. aastal, aga laialdasemalt hakati seda kasutama 2014. aastal. NodeJS on kirjutatud C++ keeles ja on lihtsasti kasutatav ning kergesti kohanduv, mis tähendab, et see võimaldab platvormist sõltumatut arendamist. Peamiselt kasutatakse seda mikroteenuste ja IoT toodete arendamiseks. NodeJS on JavaScripti *runtime environment*, mis võimaldab Javascripti koodi käivitada ka väljaspool veebibrauserit. [8]

NodeJS-i olulised eelised on kiirus ning võimalus programmeerida kliendi- ja serveripoolselt kasutades JavaScripti. Koodi käivitamise aeg on kiireim kõigist teistest programmeerimiskeeltest ning lisaks on võimalus kasutada *Node Package Manager*'i, mis pakub arendajatele mitmeid erinevaid kasulikke tööriistu ja moduleid. NodeJS arendajatel on ligipääs *Google Chrome Developer Tools*'idele, mis aitab leida vigu *backend* rakenduste koodis sama lihtsalt kui kliendirakendustes, mis jooksevad veebibrauseris. [9]

5.1.4 ExpressJS

ExpressJS on raamistik, mis töötab eelpoolmainitud NodeJS-i abil ja lihtsustab serveripoolset arendust. See on kerge ülesehitusega ja sisaldab erinevaid võimalusi ja kasulikke lisasid, mis teevad päringutega toimetamise lihtsamaks.

Antud raamistik sisaldab peamisi veebirakenduses ringi liikumiseks vajalikke funktsioone, seega arendajal ei ole vaja neid ise välja nuputada. Veebilehe struktuuri ja vajalike API-de ehitamine on selle raamistikuga tehtud lihtsaks. ExpressJS-iga saab arendada nii mobiili kui ka veebirakendusi ja selleks vajalikke API-sid. Seda on võimalik kasutada erinevate operatsioonisüsteemidega, seega tegemist on väga multifunktsionaalse ja kasuliku raamistikuga. [8]

5.1.5 MongoDB

Käesoleva töö raames loodud prototüübi arendamiseks kasutatakse MongoDB-d, mis on dokument-orienteeritud ehk NoSQL andmebaasisüsteem. NoSQL andmebaasid ei kasuta tavalisi olemisuhteid, mida kasutavad relatsioonilised andmebaasid ja enamasti ka salvestavad andmeid teises formaadis. See annab võimaluse töötada teatud hetkedel kiiremini ning lihtsamini.

MongoDB kasutab andmete salvestamiseks kollekttsioone, dokumente ning võti-väärtus paare. Andmebaasisüsteemi on võimalik kasutada erinevate operatsioonisüsteemidega ning andmeid hoitakse enamasti JSON formaadis. Antud töö raames valminud API salvestab andmeid samuti JSON formaadis, seega on andmete lugemine ja salvestamine andmebaasi kordades kiirem kui kasutades relatsioonilist andmebaasisüsteemi. [8]

5.1.6 REST API

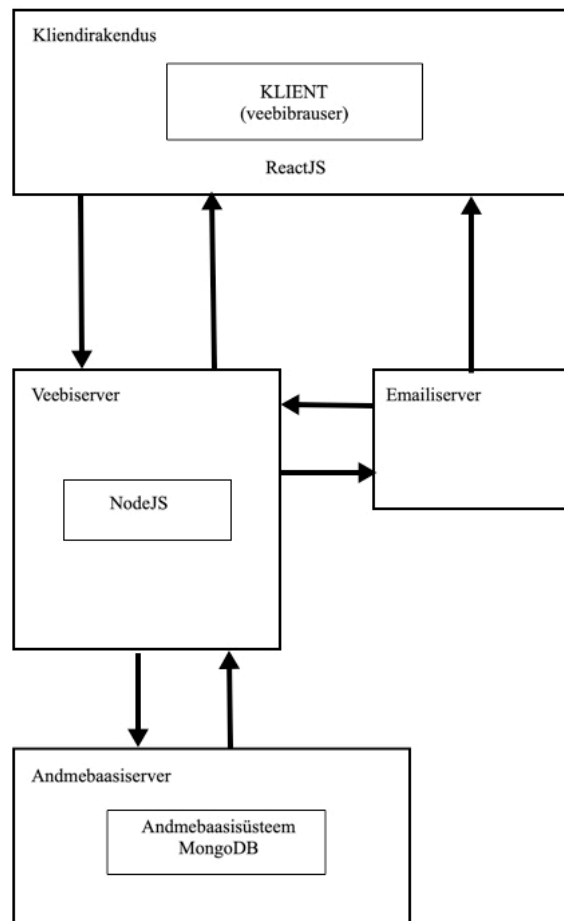
Käesoleva töö raames valmiva prototüübi arendamisel loodi RESTful API, mis kasutab eelpool selgitatud tehnoloogiaid. REST on olekuta klient-server kommunikatsiooni protokoll, mis kõikide CRUD (*Create, Read, Update, Delete*) operatsioonide puhul kasutab HTTP päringuid ja saadab enamasti vastuseid XML või JSON formaadis. Antud bakalaureusetöö raames loodud API küsib ja tagastab andmeid JSON formaadis. [10]

RESTful veebiteenuse peamised põhimõtted on järgmised:

- URL-id identifitseerivad loodud ressursi, seega need peavad olema arusaadavad ja inimloetavad
- HTTP meetodid peavad olema kooskõlas HTTP/1.1 standardiga
- API vastused ei tohi sisaldada väärtusi võtmesõnades
- Vea saamisel peab tagastatav vastus sisaldama õiget HTTP veakoodi ja infot, mille tõttu antud viga tekkis

5.2 Rakenduse arhitektuur

Kohale registreerimise rakenduse prototüübi arhitektuur koosneb järgmistest komponentidest: kliendiserver, rakendusserver, kliendi emailiserver ja andmebaasiserver.



Joonis 3. Rakenduse arhitektuur

Käivitades serveri hakkab protsess korduma vaikumisi iga tunni aja tagant, seda saab muuta vastavalt meditsiiniastutuse nõudmistele. Andmebaasist otsitakse käesoleva päeva visiidid, kellele ei ole veel infot edastatud ning saadetakse vastavale patsiendile email koos vajaliku infoga, mille põhjal saab läbida kohale registreerimise protsessi. Emaili saatmisel tehakse selle kohta ka märke andmebaasi, et vältida korduvate emailide saatmist.

Kui patsient on emaili kätte saanud ja vajutab saadud lingile, siis avaneb veebibrauseris kliendirakendus, mis teeb HTTP GET päringu veebiserverile ja jääb vastust ootama. Veebiserver pärib vastava visiidi andmeid andmebaasist visiidi identifikaatori alusel ning tagastab JSON formaadis saadud andmed kliendirakendusele. Vastavalt saadud vastusele kuvatakse kliendile, kas kehtiva visiidi broneeringu kohale registreerimise või möödunud visiidi teavituse vaade.

Kui visiidi broneering on kehtiv ja kasutaja registreerib end kohale vajutades vastavat nuppu, siis teeb kliendirakendus uue HTTP POST päringu veebirakendusele, saates JSON formaadis info, et patsient on kohale registreerinud. Kohale jõudmise registreerimist võimaldav nupp on aktiivne vaid 15 minutit enne visiidi algusaega. Veebirakendus salvestab andmed andmebaasi ning peaks saatma ka vastuse loodud API kaudu arsti rakendusele, mis teavitab, et patsient on kohale registreeritud.

5.3 API kirjeldus

Käesoleva töö raames valminud rakenduse prototüüp kasutab loodud API-t, millel on kaks ressursi. Päringute jaoks kasutatakse URL-i pealt saadud ID-parameetrit ning tagastatakse soovitud info JSON formaadis.

Patsiendi visiidi info kuvamiseks saadab veebibrauser HTTP GET päringu järgmisele URL-ile:

- `https://{domain}/api/patient/{id}`

Visiidi pärimisel süsteemist tagastatava info näidis JSON formaadis:

```
{
  "id": "5ce1c158da3cae5d84de2e96",
  "date": "2019-05-20",
  "time": "12:30",
  "doctor": "Mari Mänd",
  "identification": "6021311****",
  "email": "suuremae@gmail.com"
}
```

Patsiendi kohale registreerimiseks saadab veebibrauser HTTP PUT päringu URL-ile:

- `https://{domain}/api/update/{id}`

Päring esmalt muudab andmebaasis URL parameetrina saadud ID-ga visiidi staatust ja tagastab kohale registreeritud visiidi identifikaatori ja kinnitussõnumi, et kohale registreerimine on olnud edukas.

Kohale registreerimisel süsteemist tagastatav näidis JSON formaadis objekt:

```
{
  "id": "5ce3034fac585d6f3a7c607c",
  "message": "Patient check-in successful"
}
```

5.3.1 Emaili vaade

Kohale registreerimise protsess patsiendi jaoks algab saadud emailist. Email sisaldab linki, mis juhatab kasutaja vastavale veebirakendusele. Joonisel 4 on kuvatud saadetud emaili näidis, millelt on näha, et patsiendile ei paljastata lingis peituvad unikaalset identifikaatorit, vaid näidatakse inimkeeli mõistetava lingina.

Lugupeetud, patsient!

Olete broneerinud visiidi meditsiinasutusse X

Tervisekeskusesse kohale jõudes vajutage allolevale lingile.

[Link kohale registreerimiseks](#)

Kõike parimat!

Meditsiinasutus X

Joonis 4. Serveri saadetud emaili näidis

5.3.2 Kohale registreerimise vaated

Patsiendile saadetud email sisaldab unikaalse identifikaatoriga URL-i, mis avab veebibrauseris kliendirakenduse. Vastavalt URL-i ID parameetrile küsib rakendus API-lt visiidi andmeid, seejärel kontrollib, kas visiidi kuupäev ja kellaeg on möödunud. Kui visiidi broneering on kehtiv, siis kuvatakse kasutajale Joonisel 5 näidatud vaade.

Meditsiiniuasutus X

REGISTREERI END KOHALE

Email: kaspar.kask@gmail.com

Isikukood: 4970527****

Visiidi aeg: 19.04.2019 kell 23:00

Arsti nimi: Mari Pärn

REGISTREERI

Joonis 5. Kehtiva broneeringu kohale registreerimise vaade

Juhul kui visiidi aeg on möödunud ja seega on patsient kohale registreerimisega hiljaks jäänud, siis on protsess lõppenud ebaõnnestunult. Joonisel 6 kuvatakse antud olukorra ekraanivaade.

Meditsiiniuasutus X

REGISTREERI END KOHALE

Email: kaspar.kask@gmail.com

Isikukood: [4970527****](#)

Visiidi aeg: [19.04.2019 kell 23:00](#)

Arsti nimi: [Mari Pärn](#)

Teie visiit on kahjuks möödunud, uue aja saamiseks
pöörduge meditsiiniuasutuse registratuuri.

Joonis 6. Möödunud visiidi teavitatus

5.3.3 Visiidi toimumise info vaade

Eeldusel, et visiidi broneering on kehtiv, kuvatakse patsiendile Joonisel 5 näidatud rakenduse vaade ning patsiendil on võimalik end kohale registreerida. Registreerimise nupu vajutamisel teeb rakendus HTTP PUT päringu serverile, et muuta süsteemis visiidi staatust. Patsiendile kuvatakse Joonisel 7 näidatud vaade, kus on kirjas vajalik info toimuva visiidi kohta.

Meditšiiniasutus X

Olete registreeritud visiidile

Teid võtab vastu arst Mari Pärm

Visiit toimub kabinetis X

Joonis 7. Visiidi toimumise info vaade

5.4 Hinnang rakendusele ja nõuete täitmisele

Antud rakenduse prototüüpi anti testida autori tuttava tervisekeskuse klientidele ja töötajatele. Tagasisideks anti, et idee oma seadmest kohale registreerida on suurepärane ja tundub lihtsustavat registratuuriõdede tööd. Kui seda saaks rakendada ka suuremates haiglates, siis väldiks palju järjekorras ootamist ja säästaks aega. Mainiti ka, et veebilehe disain on lihtne ja arusaadav.

Rakendus võimaldab pärida ja seejärel kuvada nii patsiendi kui ka visiidi andmeid unikaalse identifikaatori alusel. Loodud prototüüprakenduses on võimalik patsiendil end kohale registreerida ning süsteemis vastava visiidi staatust muuta nagu nõuetes kirjeldatud.

Mingil tasemel võimaldab loodud rakendus ka isikutuvastamist ja kuna võimalus, et andmebaasis on sisestatud vigane email või keegi teine saab ligipääsu patsiendi emailile on väike, siis piisab isikutuvastamisest unikaalse URL-i põhjal, mis genereeritakse igale patsiendile visiidi ID põhjal. Nõuetekohaselt genereeritud URL saadetakse koos informeeriva emailiga patsiendile.

Prototüüp ei võimalda iga tunni aja tagant kontrollimist, kas käesolevasse päeva on broneeringuid juurde tulnud, et ka neile kohale registreerimiseks vajaliku infot edastada. See on funktsioon, mille saab projekti edasiarendamise käigus lisada.

Kõige keerukam oli prototüüpi realiseerides veebiserveri ühendamine andmebaasiga ning sinna kirjade lisamine ja kustutamine. Tegemata jäi ka veebirakenduse ja loodud teenuse ühendus arsti seadmes olema rakendusega, et kohale registreerimisel edastataks arstile vastav sõnum.

Valminud rakendust saab edasi arendada, ühendades serverit mitme erineva meditsiinasutusega. Kuna 2019. aasta alguses käivitati ka üleriigiline digiregistratuuri platvorm, siis on tõenäoline, et tulevikus hakkab sarnane rakendus töötama liidestatud digiregistratuuri külge. See võimaldaks kohale registreerimist kõikidele Eesti meditsiiniga seotud broneeringutele.

6 Kokkuvõte

Käesoleva bakalaureusetöö põhieesmärk oli analüüsida meditsiinasutusse kohale jõudmise registreerimise protsessi hetkeseisu ja vajadusi ning realiseerida kogutud info põhjal iseteeninduslik kohale registreerimise rakendus. Nagu töös selgus on meditsiinivaldkonnas enamasti pikad ootejärjekorrad tööjõu puudumise tõttu ning seega on tekkinud vajadus lihtsamaid protsesse automatiseerida.

Hetkeolukorra analüüsi tulemusena selgus, et kohale registreerimiseks kulub liialt aega ning ressursse ning tegemist on lihtsasti toimiva protsessiga, järelikul on iseteenindusliku rakenduse loomine meditsiinasutustele ärioliselt kasulik. Avaliku info põhjal tehtud analüüsi tulemusel selgus, et nii mõneski teises valdkonnas on kohale registreerimine lahendatud iseteenindusena, mis säästab nii kliendi aega kui ka ettevõtte ressursse.

Käesoleva töö raames valmis prototüüp-rakendus, mille funktsionaalsus realiseerib töö analüüsis püstitatud nõudeid ning täidab sõnastatud kohale registreerimise protsessi infovajadused. Loodud teenuse struktuur võimaldab laiendada olemasolevat funktsionaalsust, et arendada üldine rakendus, mis rahuldaks ka teisi meditsiinasutuste äriprotsesside vajadusi. Lisaks on võimalus liidestada sarnane teenus uue digiregistratuuri tarkvaraga, mis annaks võimaluse kohale jõudmise registreerimise protsessi automatiseerida ka suuremates haiglates.

Lõputöö andis autorile suurel määral uusi teadmisi veebirakenduste tehnoloogiast ning tarkvaraarenduse üldiste põhimõtetest. Kuna töös lahendatud probleem on realselt eksisteeriv, on käesolev bakalaureusetöö potentsiaalne alusdokument tervikliku ärilisel eesmärgil realiseeritava rakenduse arendamiseks.

Kasutatud kirjandus

- [1] M. L. Meuter, A. L. Ostrom, R. I. Roundtree, ja M. J. Bitner, „Self-Service Technologies: Understanding Customer Satisfaction with Technology-Based Service Encounters“, *Journal of Marketing*, kd 64, nr 3, lk 50–64, juuli 2000.
- [2] C. Williamson, „Electronic self check-in for patients: the costs and consequences“, *Br J Gen Pract*, kd 66, nr 644, lk 145
- [3] J. I. Castillo-Manzano ja L. López-Valpuesta, „Check-in services and passenger behaviour: Self service technologies in airport systems“, *Computers in Human Behavior*, kd 29, nr 6, lk 2431–2437
- [4] „The Good and the Bad of JavaScript Full Stack Development“, *AltexSoft*. .
- [5] P. Engineering, „Node.js at PayPal“, *PayPal Engineering*
- [6] Pandit, Nintin, „What Is ReactJS and Why Should We Use It?“
- [7] J. Nelson, „React explaining, explained“.
- [8] „The Good and the Bad of Node.js Web App Development“. (5.05.2019)
- [9] „6 Main Reasons Why Node.js Has Become a Standard for Enterprise-Level Organizations“. (12.05.2019).
- [10] M. Masse, *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. O'Reilly Media, Inc., 2011.
- [11] „The Good and the Bad of Node.js Web App Development“. (10.05.2019)
- [12] J. Conallen, „Modeling Web application architectures with UML“, *Communications of the ACM*, kd 42, nr 10, lk 63–70, 1999.