

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Rannar Randmäe 206021IACB

LED-tabloo täiendamine juhtmevabaks ühenduseks nutiseadmega

Bakalaureusetöö

Juhendaja: Peeter Ellervee
PhD

Tallinn 2024

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: Rannar Randmäe

20.04.2024

Annotatsioon

Käesolev lõputöö käsitleb olemasolevale LED-tabloole mobiilirakenduse loomist, mis võimaldab juhtmevaba ühenduse abil kuvada tablool teksti, visuaale ning animatsioone. Selle saavutamiseks ühendatakse LED-tablo mikrokontrolleriga ning kirjutatakse vastav tarkvara, et võimaldada Bluetooth-ühendus ja -andmeedastus nutiseadmelt LED-tabloole.

Lõputöö eesmärgi saavutamiseks olid peamisteks töövahenditeks tarkvara poole pealt C++ programmeerimiskeel, VSCode ja Xcode arenduskeskkonnad, React Native raamistik ning GitHub veebimajutus versioonihalduseks. Riistvara poolelt kasutati WS2812B LED-tablood ning ESP32-S3 mikrokontrollerit. Mikrokontrollerile korpuse loomiseks kasutati Fusion360 modelleerimistarkvara ning 3D-printerit. Lõputöö tulemusel valmis kaheosaline süsteem - volutoitel mikrokontrolleriga ühendatud LED-tablo ja React Native baasil iOS mobiilirakendus sellel tablool kuvatava sisu juhtmevabaks muutmiseks.

Lõputöö peatükkides kirjeldatakse täpsemalt süsteemi osadele seatud nõudeid, tutvustatakse LED-tablo toimimist, põhjendatakse kasutatud riistvara ja tarkvara valikut ning kirjeldatakse põhjalikult arendusprotsessi ja selle käigus valminud süsteemi.

Kogu lõputöö käigus loodud tarkvara on avatud lähtekoodiga ning avaldatud GNU GPLv3 litsensi alusel. Lähtekood on saadaval veebiaadressil <https://github.com/star2k1/ledmaster>.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 54 leheküljel, 11 peatükki, 22 joonist.

Abstract

Adapting LED Matrix Panel for Wireless Interaction with a Mobile Device

This thesis explores the creation of a mobile application designed for an existing LED matrix panel, facilitating wireless presentation of text, images and animations. The project involves integrating the LED matrix panel with a microcontroller and developing software to enable Bluetooth connectivity and data transfer between a mobile device and the LED matrix panel.

To achieve the goal of the thesis the main software tools used were the C++ programming language, VSCode and Xcode integrated development environments, the React Native framework and GitHub web hosting for version control. On the hardware side a WS2812B LED matrix panel and an ESP32-S3 microcontroller were utilized. Fusion360 modeling software and a 3D printer were used to create the enclosure for the microcontroller. The outcome of the thesis was a two-part system comprising a LED matrix panel connected to the microcontroller and a React Native-based iOS mobile application for wireless content management on the panel.

The chapters of the thesis provide a detailed description of the requirements set for the system components and introduce the operation of the LED matrix panel. The selection of hardware and software is justified, and the development process, along with the resulting system and all its components, is thoroughly described.

All software developed throughout the thesis is open-source and released under the GNU GPLv3 license. The source code is accessible at <https://github.com/star2k1/ledmaster>.

The thesis is in Estonian and contains 54 pages of text, 11 chapters, 22 figures.

Lühendite ja mõistete sõnastik

CSS	<i>Cascading Style Sheets</i> , kujunduskeel
GATT	<i>Generic Attribute Profile</i> , Bluetooth LE protokoll kiht
HEX-kood	Kuuekohaline kood, mis määrab ära iga värvi punase, rohelise ja sinise osavärvi intensiivsuse
IDE	<i>Integrated Development Environment</i> , integreeritud arenduskeskkond
JavaScript	Objektorienteeritud programmeerimiskeel
JSON	<i>JavaScript Object Notation</i> , lihtsasti loetav andmevorming
LCD	<i>Liquid Crystal Display</i> , vedelkristallkuvar
LE	<i>Low Energy</i> , madal energiatarve
LED	<i>Light Emitting Diode</i> , valgusdiod
LED-tabloo	Valgusdiodidest moodustatud kuvaseade
HTML	<i>HyperText Markup Language</i> , markeerimiskeel
Modaalaken	Ekraanipildil uue kihina avanev aken, mis suletakse pärast mingi toimingu sooritamist või info saamist
<i>Multiplexing</i>	Elektrooniline protsess, mis võimaldab ühe ühenduse abil edastada mitut elektrisignaali
React	Facebooki loodud JavaScripti veebiraamistik
React Native	Platvormist sõltumatu raamistik Reacti põhjal
RGB	<i>Red Green Blue</i> , punane, roheline, sinine
<i>Scan rate</i>	Suhtarv, mis näitab kui suur osa pikslitest LED-tabloos ühel ajahetkel põleb
SVG	<i>Scalable Vector Graphics</i> , vektorgraafika vorming
Sõne	Andmetüüp programmeerimiskeeltes (<i>string</i>)
Teek	Funktsioonide, moodulite või muude komponentide kogu, mida saab programmis vajadust mööda kasutada
UX	<i>User experience</i> , kasutajakogemus
Välkmälu	Säilmälu, milles andmed säilivad ka toite väljalülitamise järel

Sisukord

1 Sissejuhatus	9
2 Nõuded süsteemi osadele.....	10
2.1 Rakenduse nõuded	10
2.2 LED-tabloo nõuded	12
2.3 Mikrokontrolleri nõuded.....	12
3 LED-tabloodest.....	14
4 Kasutatav riistvara	17
4.1 WS2812B LED-tabloo	17
4.2 ESP32-S3 mikrokontroller	18
5 Kasutatav tarkvara	20
5.1 Arenduskeskkonnad.....	20
5.1.1 Arduino IDE	20
5.1.2 VSCode ja PlatformIO	20
5.1.3 Xcode.....	21
5.2 Fusion360	22
5.3 Versioonihaldus	22
5.4 Teegid	22
6 Tööprotsess.....	23
6.1 Ettevalmistus	23
6.2 Mobiilirakenduse arendus.....	24
6.2.1 Tehnoloogia valik.....	24
6.2.2 Kujunduse valik.....	25
6.2.3 Protsess	26
6.3 Sardtarkvara arendus	28
6.4 Viimistlus.....	30
7 Sardtarkvara.....	32
7.1 Bluetooth-ühendus.....	32
7.2 Juhtloogika.....	33
7.2.1 Alglaadimine	34

7.3 Klassid ja nende funktsioonid.....	35
7.4 Eraldiseisvad funktsioonid	36
8 Mobiilirakendus.....	39
8.1 Kasutusjuhud	39
8.2 Arhitektuur.....	44
8.3 Ladu	44
8.4 Komponendid	45
8.4.1 Külgmenüü	45
8.4.2 Kujunduse eelvaade.....	46
8.4.3 Lõuend.....	47
8.4.4 Keelevaliku modaalaken	48
8.4.5 Ühenduste modaalaken.....	49
8.4.6 Värvivaliku modaalaken.....	50
8.5 Teenused.....	50
8.5.1 Keeleteenus.....	50
8.5.2 Teavitusteenus	50
8.5.3 hexToBitmap	51
8.6 Kuvad.....	52
8.6.1 Avaleht	52
8.6.2 Eelloodud kujundused	54
8.6.3 Minu kujundused	55
8.6.4 Tekst	56
8.6.5 Loo visuaal	57
8.6.6 Loo animatsioon	57
8.6.7 Seaded.....	58
9 Lõpplahendus	59
10 Võimalikud edasiarendused.....	60
11 Kokkuvõte	62
Kasutatud kirjandus	63
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks	68
Lisa 2 – Sardtarkvara juhtloogika lihtsustatud voodiagramm.....	69
Lisa 3 – Mobiilirakenduse komponentdiagramm.....	70
Lisa 4 – Lõpplahenduse elektriskeem	71

Jooniste loetelu

Joonis 1. Mobiilirakenduse „CoolLED1248“ katkine kasutajaliides ning piiratud värvivalik.....	11
Joonis 2. Valik erinevaid LED-tabloosid [12].....	16
Joonis 3. Fusion360 modelleerimistarkvaras loodud arendusplaadi korpus.	30
Joonis 4. Fusion360 modelleerimistarkvaras loodud arendusplaadi korpus vol. 2.	31
Joonis 5. Diagramm Bluetooth LE GATT kihi kasutusest loodud süsteemis.	33
Joonis 6. Sardtarkvara juhtloogika üldistatud voodiagramm.	34
Joonis 7. Sardtarkvara alglaadimise lihtsustatud voodiagramm.....	35
Joonis 8. Mobiilirakenduse kasutusjuhtude diagramm.	40
Joonis 9. Mobiilirakenduse lao osad ning osades paiknevad väärtused ja funktsioonid.	45
Joonis 10. Mobiilirakenduse külgmenüü komponent.....	46
Joonis 11. Mobiilirakenduse kujunduse eelvaate komponendi erinevad kasutused.....	47
Joonis 12. Mobiilirakenduse lõuendi komponent LED-tabloole vastava kuvasuhte ning pikslite arvuga.	48
Joonis 13. Mobiilirakenduse keeleveliku modaalaken.....	49
Joonis 14. Mobiilirakenduse ühenduste modaalaken.	49
Joonis 15. Canva keskkonnas loodud maketid mobiilirakenduse kuvadele.....	52
Joonis 16. Mobiilirakenduse avaleht ning lähivaade keeleveliku nupule.	53
Joonis 17. Mobiilirakenduse kuva „Eelloodud kujundused“ visuaalide sektsioon.	54
Joonis 18. Mobiilirakenduse kuva „Minu kujundused“ visuaalide sektsioon.....	55
Joonis 19. Mobiilirakenduse kuva „Tekst“.	56
Joonis 20. Mobiilirakenduse kuva „Loo visuaal“.....	57
Joonis 21. Mobiilirakenduse kuva „Loo animatsioon“.	58
Joonis 22. Lõpplahenduse üldine skeem.	59

1 Sissejuhatus

Käesoleva lõputöö eesmärk on luua lahendus, mis võimaldaks juhtmevabalt kuvada LED-tablool kasutaja poolt loodud visuaale või sõnumeid. See idee tekkis autorile Egiptuses reisil olles, kus ta nägi hulgaliselt LED-tabloosid, mida kasutati erinevatel otstarvetel. Need olid poekendel, kus need kuvasid tervitavaid sõnumeid, autoklaasidel, teavitades, et tegemist on taksoga, või kuvades mõnda reklaami, aga ka liiklusmärkide asemel, kuvades staatilise juhise asemel hoopis paremini tähelepanu äratavat animatsiooni. Sellest tulenes ka autori soov näha kodumaal enda ümber rohkem LED-tabloosid.

Üks viis selle saavutamiseks on luua vabalt kättesaadav kaasaegne eestikeelne mobiilirakendus, mis võimaldaks igäihel taolisi tabloosid meelepärasel viisil kasutada. Hetkel turult leitavad tooted ei olnud autori hinnangul taskukohased, olid liiga väheste võimalustega, ei võimaldanud head mobiilsust, või nende mobiilirakendus ei pakkunud autorile meeldivat kasutuskogemust. Lisaks autori isiklikule huvile LED-tabloode vastu on valitud teema ka aktuaalne, kuna LCD (*Liquid Crystal Display*) ekraanide kasutus on languses ning erinevaid LED-tehnoloogiaid peetakse ekraanide tulevikuks [1].

Antud lahenduse realiseerimiseks oli vaja leida sobiv LED-tabloo, luua selle juhtimiseks mobiilirakendus, valida välja sobiv mikrokontroller ja programmeerida see vastavalt, et see oleks vahelülis tabloo ja mobiilirakenduse vahel. Lisaeesmärk, mida proovitakse saavutada, on loodud prototüübi visuaalse ilme parandamine ning komplektseks tegemine, et seda oleks võimalik mugavalt paigaldada ja kasutada.

Lõputöö käigus defineeritakse esmalt nõuded loodava rakenduse ning prototüübi jaoks, seejärel uuritakse LED-tabloode toimimist, soetatakse sobiv riistvara ning alustatakse sardtarkvara ja mobiilirakenduse arendusega kasutades erinevaid programmeerimiskeeli ja raamistikke. Arendusprotsessi vältel kasutatakse versioonihaldust, keskendutakse hea kasutajakogemuse saavutamisele ning juhitudakse puhta koodi põhimõtetest. Tutvutakse ka modelleerimisega, et luua prototüübi ilme parandamiseks kasutatavale arendusplaadile korpus.

2 Nõuded süsteemi osadele

Järgnevad alapeatükid kirjeldavad täpsemalt lõputöö käigus valmivale süsteemile seatud nõudeid. Igas alapeatükis sõnastatakse erineva süsteemiosa nõuded ning tuuakse välja ka põhjendused.

2.1 Rakenduse nõuded

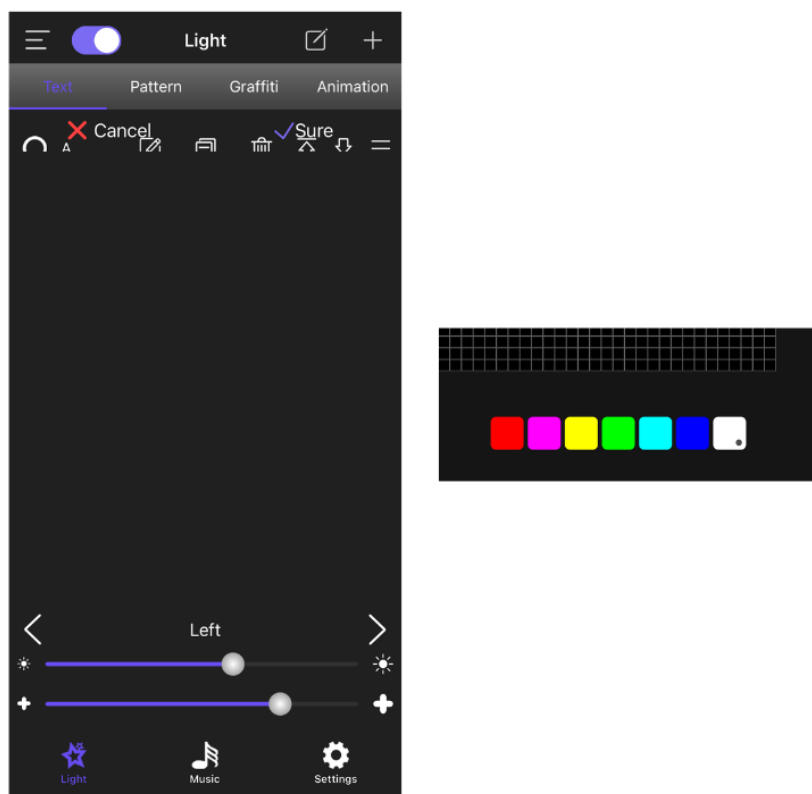
Mobiilirakenduse loomisel lähtub autor puhta koodi põhimõtetest, kuna see aitab koodi muuta loetavamaks ka teistele, ning kasutab arenduseks ajakohaseid ja tasuta kättesaadavaid tööriistu [2]. Rakenduse kasutatavad teegid peavad olema avatud lähtekoodiga.

Seadme valikul, millele rakendus luuakse, lähtus autor omale kättesaadavatest juhtmevaba ühenduse võimekusega seadmetest, milleks olid sülearvuti ja nutitelefon. Lisaks võttis autor arvesse, millisest seadmest tal kõige mugavam loodavat lahendust kasutada oleks. Kuna loodava LED-tabloo eeldatavad kasutuskohad on autoklaas, kaupluse aken või mõne riidetuse osa, jõudis autor järeldusele, et kõige mugavam oleks LED-tablood juhtida nutitelefoni abil. Nutitelefon on enamik inimestel kogu aeg käepärast, vahet pole millise valdkonnaga nad tegelevad. Autoril on ligipääs vaid Apple'i nutiseadmetele, seega toimub ka arendus eelkõige nende jaoks ehk loodav rakendus peab jooksma iOS operatsioonisüsteemis.

Vaadates olemasolevaid lahendusi, leidis autor, et üks eelnevaid nõudeid täitev rakendus nimega WLED (*WiFi Lighting Effects Driver*) leidub, kuid see kasutab LED-tabloo ja nutiseadme vaheliseks ühenduseks internetti [3]. Autori soov on, rakenduse kasutamine ei sõltuks internetiühenduse olemasolust. See tähendab, et WiFi kaudu LED-tabloo ühendamine ja WLED on välistatud ning peab kasutama teist tänapäeva seadmetes saadavalolevat juhtmevaba ühenduse varianti, milleks on Bluetooth. Bluetooth'i on kahte varianti: klassikaline ja Bluetooth *Low Energy* (edaspidi: Bluetooth LE). Klassikalist Bluetooth'i kasutatakse peamiselt suuremahuliseks ja pidevaks andmeedastuseks, näiteks juhtmevabades kõrvaklappides, ning see tarbib ka rohkem voolu. Antud rakenduse puhul

sobib paremini madala energiatarbega Bluetooth'i variant, kuna andmeid tuleb LED-tabloole saata vaid vahetevahel ning see võimaldab säästa ka nutiseadme akut. Ainuke mööndus Bluetooth LE puhul on see, et seda kasutavaid seadmeid ei leia otse nutiseadme seadetest Bluetooth-ühenduste valikust, vaid see funktsionaalsus tuleb rakenduses ise implementeerida [4], [5].

Leiduvad ka mõned rakendused, mis ongi mõeldud teatud LED-tabloodega ühendamiseks Bluetooth'i abil, kuid autor leidis, et need võimaldavad juhtida ainult valitud Hiinast tellitavaid tooteid [6]. Lisaks olid nendel rakendustel muud miinused, nagu katkine ja vananenud kasutajaliides, piiratud värvivalik LED-tablool sisu kuvamiseks, väikene seadistamisvõimalus ja eesti keele puudumine (vt Joonis 1).



Joonis 1. Mobiilirakenduse „CoolLED1248“ katkine kasutajaliides ning piiratud värvivalik.

Kuna leitud rakendustel puudus avatud lähtekood, ei saa kasutajad soovi korral rakendusele ka ise uusi funktsionaalsusi või mitte Hiina päritolu LED-tabloode jaoks tuge lisada. Nendest asjaoludest lähtuvalt otsustas autor, et lõputöö käigus loodaval rakendusel peab olema avatud lähtekood, et kogukond saaks lisada rakendusele funktsionaalsusi, aidata leida ning parandada vigu ning lisada tuge veel toetamata LED-tabloodele. Et

rakendus oleks kasutatav suuremale arvule kasutajatele, peaks seal olema ka võimalus valida mitme keele vahel.

Loodav rakendus peab kasutajal võimaldama kontrollida LED-tablood järgnevatel viisidel: sisse- ja väljalülitamine, heleduse muutmine, sõnumite, visuaalide ja animatsioonide tabloole saatmine. Rakenduses peab olema võimalus valida saadetak visuaal olemasolevate seast, luua ise visuaal, sõnum või animatsioon ning muuta varasemalt loodud visuaale.

2.2 LED-tabloo nõuded

Mitmeotstarbeliseks kasutuseks valmiva lahenduse LED-tabloo peab võimaldama kuvada teksti, pilte ning animatsioone, et seda saaks kasutada nii reklaamiks, meelelahutuseks kui ka informatsiooni edastamiseks. Tabloo peab võimaldama kuvada sisu nii ühe- kui ka mitmevärviliselt. LED-tabloo peab võimaldama iga üksiku piksli väljalülitamist või värvi muutmist. See on oluline, kuna väiksemate LED-tabloode madala resolutsiooni puhul on iga piksel silmaga eristatav. Juhtmevaba ühenduse realiseerimiseks peab LED-tablood saama ühendada mikrokontrolleriga, kuhu on võimalik saata käsklusi selle juhtimiseks. See tähendab, et LED-tablool peaks olema pistik või lihtsasti ligipääsetavad väljaviigud. LED-tabloo peab olema piisavalt õhuke või painduv, et seda oleks võimalik mugavalt kaasas kanda ning kinnitada ka kumeratele pindadele. Prototüübi staadiumis peab LED-tabloo toimima vaid siseruumides, seega veekindlus ega temperatuuritaluvus pole antud lõputöö skoobis. Lahenduse tabloo peaks olema taskukohase hinnaga ning laialtlevinud, et arendusprotsessi käigus toote purunemise korral oleks võimalik uus tellida, et leiduks abimaterjale ning lõpphind kujuneks võimalikult soodne.

2.3 Mikrokontrolleri nõuded

Antud lahenduse realiseerimiseks vajaliku mikrokontrolleri peamised nõuded on sisseehitatud Bluetooth-moodul ja piisav võimekus samaaegselt LED-tablood juhtida ning Bluetooth-ühendusega tegeleda. See tähendab seda, et mikrokontroller võiks omada kahte tuuma, millest üks tegeleb seadmetevahelise ühendusega ning teine LED-tablood juhtimisega. Väiksemate LED-tabloode puhul piisaks ilmselt ka ühest tuumast, kuid programmi parema skaleeritavuse huvides valis autor kahe tuuma nõude.

Mikrokontrolleril peab olema ka piisavalt mälu, et hoiustada lisaks programmikoodile ka sellele edastatud visuaale, animatsioone või sõnumeid. Tehnoloogia arenguga kaasas käimiseks võiks mikrokontroller omada USB-C pesa, kuna Micro-USB ühendust enam tänapäeva seadmed eriti ei kasuta. Veel soovib autor, et mikrokontroller oleks taskukohase hinnaga ja laialtlevinud, et hiljem oleks tootel rohkem kasutajaid ning vajadusel leida dokumentatsiooni ja näidismaterjale.

3 LED-tabloodest

LED-tablood on saanud oluliseks info tarbmise osaks meie igapäevaelus. Eestis võib neid näha ühistranspordis marsruute ning bussiaegu kuvamas, tänavatel ning hoonetes reklaamtahvlitena ning ka liikluses elektrooniliste liiklusmärkidena. Peale informatiivse väljundi on LED-tabloodel aga ka teine eesmärk – olustiku loomine ning meelelahutus. Neid saab kasutada valgusinstallatsioonides, hoonete fassaadidel, kodudes sein- või lauakaunistusena, kuid ka hobi- või teadusprojektides. Sellise laia amplituudi tõttu võib LED-tabloosid pidada vägagi perspektiivikateks seadmeteks. Käesoleva peatüki eesmärk on anda ülevaade LED-tabloode omadustest ning toimimisest, et tekiks parem arusaam, millega autor antud lõputöös tegeleb.

LED-tabloo on LED-idest koosnev kuvaseade, mida enamasti kasutatakse erineva teabe edastamiseks. LED-id paiknevad tabloos maatriksina moodustades read ja veerud. Sellest tuleneb ka LED-tabloole ingliskeelne nimetus *LED Matrix Panel*.

LED-tabloosid juhitakse voolu abil ning kuna LED-ide arv võib olla väga suur, kasutatakse enamikel juhtudel juhtimiseks tehnikat nimega *multiplexing*. Antud tehnika kujutab endast seda, et ühel ajahetkel antakse voolu vaid ühele reale LED-idele. Sama perioodi vältel antakse voolu ka veergudele, milles paiknevad LED-id, mida soovitakse sütitada. See võimaldab iga LED-i eraldi adresseerimist, kuid nõuab ka omajagu arvutusvõimsust, mis tähendab et tuleb kasutada võimekaid kontrollereid [7]. Et vältida kõigi mikrokontrolleri väljaviikude ärakasutamist LED-tabloole poolt, kasutatakse nihkeregistreid, mis muudavad järjestiksiselt mikrokontrollerist paralleelväljundiks LED-tabloole. Traditsioonilistel LED-tabloodel on seda vaja, kuna iga värvikanali juhtimiseks on vaja mitut ühendust [8].

On ka teist tüüpi LED-tablood, mida autor kasutab selles lõputöös. Selle tüüpi puhul on igaks pikslik LED-kiip, mis võimaldab järjestikandmeedastuse kasutades vaid ühte mikrokontrolleri väljaviiku. See saavutatakse nii, et kõigi pikslite värviinfo saadetakse esimesse pikslisse, kus piksli enda signaal lahutatakse ning ülejäänud pikslite andmed saadetakse edasi järgmisesse pikslisse. Niimoodi toimub see seni, kuni info on otsas ja on

jõutud viimase pikslini. Antud tüüpi LED-tablood on küll väga mugav juhtida, kuid sellega seoses suureneb võrreldes *multiplexing* tehnikaga mitmekordselt ka voolutarve [9].

Üheks oluliseks LED-tablood kirjeldavaks omaduseks on pikslite vahe või *pixel pitch* – see tähistab kaugust kahe järjestiku piksli vahel ning seda mõõdetakse millimeetrites ühe piksli keskpunktist teise piksli keskpunktini. Piksli vahet tähistatakse tähega „P“ ehk näiteks kümne millimeetrise piksli vahe puhul oleks LED-tablool spetsifikatsioonis kirjas „P10“. Erinevate piksli kõrgustega tabloosid kasutatakse erinevate vaatamiskauguste jaoks. Lähemalt vaatamiseks, näiteks siseruumide seintel, sobivad väiksema piksli vahega tablood ja kaugemalt vaatamiseks, näiteks välireklaamid või hoonete fassaadid, suurema piksli vahega tablood. Antud omadust tuleb LED-tablool soetamisel silmas pidada, sest muidu võivad mitu sama resolutsiooniga tablood osutada suuruselt erinevaks ja seega omavahel mittesobituvaks [10].

Tüüpilistele LED-tabloodele spetsiifiline omadus on ka *scan rate*, mida näidatakse tavaliselt suhte või murdarvuna. See omadus kirjeldab, kui suur osa tablool pikslitest ühel ajahetkel põleb. Antud omadus on LED-tabloodel sellepärast, et korraga kõigi pikslite sütitamine tarbiks liiga palju voolu. Selle asemel luuakse pilt sütitades pikslid ridade kaupa ja tekib skännimise efekt. Kuna seda tehakse suurel sagedusel, siis inimsilm pole võimeline tuvastama, et tegu ei ole tervikliku pildiga. See on ka põhjus, miks videole jäänud LED-reklaamtahvli peal võib näha sellest üle jooksvaid jooni. Hobi korras kasutatavate LED-tabloode puhul on tüüpiline *scan rate* 1:16 ehk ühel ajahetkel põleb korraga 1/16 tablool pikslitest [11].

LED-tabloodel võivad olla ühe-, kahe- või mitmevärvilised. Informatiivsel eesmärgil kasutatavad tablood, näiteks ühistranspordis või spordivõistlustel on tavaliselt ühe- või kahevärvilised, et info oleks selgelt nähtav ning üheselt mõistetav. Reklaami, meelelahutuse või hobi korras kasutatavad tablood on peamiselt mitmevärvilised, et saaks kuvada võimalikult tähelepanu äratavaid visuaale (vt Joonis 2).



Joonis 2. Valik erinevaid LED-tabloosid [12].

Võrreldes LED-tablood teise tuntud ekraanitehnoloogia, LCD-ga, mida on pikalt kasutatud monitorides, telerites ja nutiseadmetes, on LED-tablood tavaliselt madalama resolutsiooniga ning pikseleid on võimalik üksteisest eristada [13]. Üha enam võetakse LED-tabloosid kasutusele ka LCD ekraanide asemel ja seda headel põhjustel. Erinevalt LCD ekraanidest ei pea LED-tabloodel modulaarsuse tõttu kindlalt fikseeritud kuju ja suurust olema – LED-tablood koosnevad tavaliselt paljudest väiksematest omavahel ühendatud moodulitest. Üldiselt on LED-tabloode eluiga kordades pikem kui LCD ekraanidel – 11 aastat versus 3-5 aastat. LED-tablood on tunduvalt eredamad tänu sellele, et nad ei kasuta taustvalgustust, vaid kiirgavad valgust otse. LED-tabloodel puudub peegeldus, mis tähendab valgustatud oludes paremat nähtavust ja pildikvaliteeti [14], [15]. Lisaks eelistele LCD ekraanide ees on LED-tabloode valdkond pidevas arengus ning peagi võivad siinsetele poeklaasidele jõuda ka kõige kaasaegsemad tehnoloogiad, mis kasutavad läbipaistvat trükkplaati [16].

4 Kasutatav riistvara

Käesoleva lõputöö raames kombineeriti olemasolevaid riistvaralisi komponente ning nullist midagi ei loodud. Uute komponentide loomisel ei olnud vajadust, kuna lõputöö eesmärk ei olnud luua kõige paremate näitajatega LED-tablood ja selle juhtimissüsteemi, vaid eelkõige luua kasutajasõbralik vabavaraline mobiilirakendus ja täiendada olemasolevat tablood vastavalt, et seda oleks võimalik nutiseadmest juhtida. Lisaks ei olnud eesmärgiks saada lõpptoodet, vaid luua prototüüp. Järgnevas kahes alapeatükis kirjeldatakse riistvaraliste komponentide valikut täpsemalt.

4.1 WS2812B LED-tablo

Esialgu oli autoril plaan kasutada 16x64 resolutsiooniga tablood, kuna see võimaldanuks kuvada rohkelt erinevaid sümboleid ning luua detailsemaid visuaale. Kahjuks ei suutnud autor esialgu ühtegi taolise resolutsiooniga lõputöös seatud nõuetele vastavat LED-tablood leida, mida saaks kasutada. Leidus küll painduvaid 16x64 tabloosid, mis olid juba eelnevalt kontrolleri ning mobiilirakendusega liidestatud ning ühe taolise ka autor Temu veebipoest tellis, kuna hind oli seal poole soodsam kui analoogsel tootel Eesti kaupluses [17]. Toote kättesaamisel aga selgus, et sellest ainult tablo komponenti eraldamine ning hiljem oma kontrolleri liidestamine osutuks liialt ajakulukaks ning keerukaks, kuna toote andmeleht pole avalikult saadav ning toote rakenduse lähtekoodile ligipääs puudub. Lisaks ei leidnud autor ka võimalust loodava rakenduse liidestamiseks antud tootega.

Eelnevatest asjaoludest tingitult tuli arenduseks valida mingi muu resolutsiooniga tablo. Heade omaduste ja nõuetele vastavuse tõttu otsustati valida neli korda väiksema resolutsiooniga 8x32 WS2812B LED-tablo. WS2812B, mida tuntakse ka kui „NeoPixel“, tähistab pindmonteeritavate LED-kiipide tüüpi, millest tablo koosneb. Tegemine on 5-millimeetrise läbimõõduga targa LED-kiibiga, kuna lisaks punasele, rohelinele ning sinisele LED-ile sisaldab see samas kestas ka juhtloogikat [9]. Valitud tabloos on LED-kiibid paigaldatud õhukesele trükkplaadile ning ühendatud järjestikku. See võimaldab tabloole natukene painduvust, kogu värviskaala kasutamist ning iga LED-i eraldi adresseerimist. Antud tablo mõlemas ääres on pistik, millest üks võimaldab saata

andmeid sisse ning teine saata andmeid edasi, kui on soov tabloosid juurde lisada. LED-idesse sisseehitatud juhtkiipide tõttu kasutatakse andmeedastuseks vaid ühte ühendust. Tabloo keskel on ühendused vooluallika jaoks [18].

Sisendpinge on tablool viis volti ning voolutarve võib maksimaalsel võimsusel ulatuda kuni 15 amprini. Voolutarve arvutatakse korrutades LED-kiipide arvu maksimaalse voolutarbega ühe LED-i kohta. Antud tabloo puhul on ühe LED-i maksimaalne voolutarve andmelehe järgi 0.02 amprit [9]. Tabloos on kokku $256 * 3 = 768$ LED-i (igas pikslis kolm erinevat värvi) ning $768 * 0.02 = 15.36$ amprit. Lõputöö raames seatakse tabloo eredusele tarkvaraline piirang, et voolutarve ei ületaks kolme amprit. Tabloo reklaamitav tööiga on enam kui 50 000 tundi [19].

Antud komponent otsustati tellida Hiina veebipoest Aliexpress.com, sest vaatamata pikemale tarneajale oli komponendi hind seal enam kui kümme korda soodsam (€8), kui analoogsel tootel Eesti poes (€109) [19], [20]. Lisaks headele omadustele saab tablood kasutada paljude teekidega ning suure kasutajaskonna tõttu leidub sellele ka mitmeid juhendeid.

Valitud LED-tablool on ka mõned puudused: tabloo on küll painduv, kuid seda ei saa painutada igas suunas ega kokku rullida; voolutarve on väikse tabloo kohta väga suur; puudub igasugune vee- ja ilmastikukindlus; puudub kate, mis LED-idest tulevat valgust pisut hajutaks. Antud lõputöö raames pole need puudused määrava tähtsusega, kuna vool võetakse seinapistikust ning tablood kasutatakse sisetingimustes.

4.2 ESP32-S3 mikrokontroller

Käesoleva lõputöö realiseerimiseks on vaja luua juhtmevaba ühendus nutiseadme ning LED-tabloo vahel. Valitud LED-tabloos on igas pikslis küll eraldi juhtloogika, kuid see ei paku juhtmevaba ühenduse funktsionaalsust, seega on vaja mingit vahelüli, et info ühest seadmest teiseni jõuaks. Antud probleemi lahenduseks peab kasutama mikrokontrollerit.

Sobiva mikrokontrolleri valikul lähtus autor mikrokontrollerile seatud nõuetest ning LED-tabloo juhtimiseks loodud teekide saadavusest antud mikrokontrollerile. Kuna lõputöö käigus valmib lahenduse prototüüp, mitte lõpptoode, otsustas autor kiirema ja mugavama arendusprotsessi huvides kasutada arendusplaati, selle asemel, et ise

mikrokontrolleri jaoks trükkplaat disainida. Küll aga on trükkplaat võimalik disainida antud lõputöö võimaliku edasiarendusena.

Ülikoolis läbitud projektide tõttu mäletas autor, et üks kõige levinuimaid ja parema võimekusega taskukohaseid mikrokontrollereid on Espressifi loodud ESP32. Esialgu oli valikus ka ESP8266, kuid peagi sai selgeks, et Bluetooth'i puudumise ning ühe tuuma tõttu ei ole see antud lõputöö jaoks piisav [21]. Esialgu ei olnud autor kursis, et ESP32 mikrokontrollerist on omakorda palju erinevaid variante ning tellis veebipoest Amazon.de ühe soodsama ESP32 arendusplaadi variandi. Selle variandi valikul pidas autor silmas USB-C ühenduse olemasolu ning heasid arvustusi. Tellitud arendusplaat kasutas ESP32-WROOM-32 moodulit, mis on üks vanemaid ESP32 variante [22]. Kahjuks selgus pärast testkoodide jooksutamist, et arendusplaadid olid vigased, või vajasis mingit lisakonfiguratsiooni, mida alustamisjuhendis kirjas ei olnud. Ühe arendusplaadi puhul ei toiminud Bluetooth LE ühendus, mida plaaniti ülesande lahendamisel kasutada. Teise arendusplaadi puhul ei toiminud isegi tavaline Bluetooth-ühendus. Sellest lähtuvalt tegi autor otsuse antud arendusplaadid tagasi saata ning uued tellida.

Uute arendusplaatide valikul oli autor juba rohkem teadlik erinevatest pakutavatest mikrokontrolleri variantidest ning valis ühe võimekama ESP32 variandi, milleks oli ESP32-S3. Antud mikrokontrolleri mõned spetsifikatsioonid on kahetuumaline 32-bitine Xtensa LX7 protsessor, 8 MB välmälu ning WiFi ja Bluetooth 5 LE mooduli olemasolu. Nendest andmetest saab välja lugeda, et mikrokontroller täidab sellele seatud nõudeid. Sobiva mikrokontrolleriga arendusplaat Waveshare ESP32-S3-DEV-KIT-N8R8 telliti Poola veebipoest botland.store, kuna seal oli kiire tarne ning taskukohane hind. Arendusplaadi valikul sai määravaks USB-C pesa, sisseehitatud RGB LED, suur väljaviikude arv ning viie voldi väljund, mida sai esialgu kasutada LED-tabloode testimiseks [23].

Arendusplaadi valikul ei peetud silmas, et see peaks sobima kasutuseks täpselt LED-tabloode jaoks, vaid valiti plaat, mida saab hiljem ka muul otstarbel ära kasutada, kui arendataval LED-tablool pärast edasiarendusi arendusplaati enam vaja ei peaks minema.

5 Kasutatav tarkvara

Käesolev peatükk tutvustab lõputöö arendusprotsessi käigus kasutatud tarkvara. Sinna hulka kuuluvad arenduskeskkonnad, teegid ja modelleerimistarkvara mikrokontrollerile korpuse loomiseks.

5.1 Arenduskeskkonnad

Antud alapeatükk tutvustab kolme erinevat arenduskeskkonda, mida käesoleva lõputöö raames kasutati. Tuuakse välja keskkondade kirjeldus ning otstarve, milleks neid kasutati.

5.1.1 Arduino IDE

Esialgu kasutati ESP32 mikrokontrollerile sardtarkvara arenduseks Arduino IDE tekstiredaktorit, kuna autor oli seda keskkonda varasemate projektide jooksul kasutanud. Arduino IDE on Arduino baasil töötavatele mikrokontrolleritele koodi kompileerimiseks ja üleslaadimiseks mõeldud arenduskeskkond. Keskkond on võrreldes paljude teiste arenduskeskkondadega tehtud väga lihtsaks, et algajal arendajal oleks võimalikult lihtne oma kood mikrokontrolleri peal jooksmata saada. Lihtsusega kaasnevad aga ka miinused: puudub mugav versioonihaldus; suurema projekti puhul on keskkonnas ebamugav mitmete erinevate failidega tegeleda; puudub koodis vigade tuvastamise võimalus enne koodi kompileerimist; puuduvad koodi kirjutamisel soovitud, et lõpetada pooleliolev koodirid; kasutusel on lihtsustatud versioon C++ keelest ning seega ei saa kõiki sisseehitatud funktsioone kasutada [24]. Seoses nende puudujääkidega otsustas autor arenduse käigus sardtarkvara arenduseks keskkonda vahetada.

5.1.2 VSCode ja PlatformIO

Enamik arendusest toimus kasutades VSCode (*Visual Studio Code*) tekstiredaktorit. Antud tarkvara kasutati järgnevatel põhjustel: varasem kogemus antud arenduskeskkonnas; suur valik laiendusi töö hõlbustamiseks; tegu on vabavaralise tarkvaraga Microsofti poolt ehk see on kättesaadav tasuta ning leidub palju tugimaterjale; hea liidestus versioonihaldustarkvaraga ning võimalus arendada ühes keskkonnas kasutades erinevaid programmeerimiskeeli [25].

PlatformIO tarkvara kasutati ESP32 mikrokontrolleri sardtarkvara koodi kompileerimiseks ning mälusse laadimiseks. PlatformIO on sarnaselt Arduino IDE'le C ja C++ keeles mikrokontrolleerite arenduseks mõeldud tarkvara ning toetab lisaks Arduino baasil töötavatele mikrokontrolleeritele veel mitmeidki teisi. PlatformIO on saadaval laiendusena VSCode'i jaoks, mis võimaldab seda kasutada ilma eraldiseisvat programmi installeerimata [26]. Antud tarkvara võeti kasutusele Arduino IDE asemel, kuna sellega on võimalik kasutada VSCode'i eeliseid: hoida nii mobiilirakenduse kui ka sardtarkvara lähtekood ühes keskkonnas ja saada koheseid soovitusi koodi parandamiseks, ilma et peaks ootama kuni kood kompileerub. See tagab kiirema ja mugavama arendusprotsessi ning ühtse versioonihalduse kogu lõputöö projekti raames. Üleminek Arduino IDE'st PlatformIO'le ei olnud keeruline, sest selleks vastav dokumentatsioon oli veebist leitav ning kõiki Arduino IDE's välja valitud teeke sai samuti kasutada ka PlatformIO's.

5.1.3 Xcode

Xcode on arendajatele mõeldud tööriistade komplekt, et luua rakendusi Apple platvormidele. Xcode tarkvara kasutati mobiilirakenduse kompileerimiseks ning seadmes testimiseks. Antud tarkvara oli vaja kasutada, kuna arendati iOS rakendust ning see on ainus võimalus, kuidas Apple lubab enda seadmetele tarkvara arendada ja avaldada. Xcode sisaldab endas nii tekstiredaktorit kui ka seadme simulaatorit. Tekstiredaktorit kasutusele ei võetud, kuna selleks otstarbeks kasutati VSCode'i. Küll aga kasutati simulaatorit, et mobiilirakendust testida [27].

Apple'i seatud piirangute tõttu on Xcode'i võimalik kasutada ainult macOS operatsioonisüsteemiga seadmetes. Windows'i või Linux operatsioonisüsteemis Xcode kasutamiseks on mõningad lahendused küll olemas, näiteks virtuaalmasinas või pilvepõhiselt macOS jooksutamine, kuid need lahendused on tasulised või siis rikuvad Apple'i poolt sätestatud Xcode'i kasutamise nõudeid [28]. Kuna autoril isiklik macOS operatsioonisüsteemiga seade puudus, kuid lõputöö vajas tegemist, pöördus ta lahenduse leidmiseks juhendaja poole. Juhendaja abiga sai autor ligipääsu IT Kolledži arvutiklassile ICO-320, kus olid kasutada iMac arvutid, mis olid oma võimekuse tõttu arenduseks ideaalsed.

5.2 Fusion360

Fusion360 on Autodesk'i poolt loodud modelleerimistarkvara, mida autor kasutas mikrokontrolleri korpuse 3D-prinditava korpuse mudeli loomiseks. Autoril varasem kogemus 3D-modelleerimisega puudus, kuid internetis leitavate juhendite abil loodi kahes iteratsioonis prototüübi jaoks sobiv korpus. Fusion360 tarkvara valiti seetõttu, et see oli tasuta kättesaadav ning juhendis, mis õpetas ESP32 arendusplaadi korpuse modelleerimist, kasutati just seda [29], [30].

5.3 Versioonihaldus

Versioonihalduse jaoks kasutas autor Git'i ja GitHub'i, kuna tal oli nendega olemas varasem kogemus ja need on üldiselt sarnaste arenduste puhul standardiks. Lisaks on Git'il VSCode laiendus, mis tegi selle kasutamise veelgi mugavamaks. Kuna käesoleva lõputöö eesmärk oli luua avatud lähtekoodiga rakendus, on kogu kood saadaval avalikus GitHub'i hoidlas.

5.4 Teegid

Mikrokontrolleri sardtarkvara poolt kasutatud kolmandate osapoolte kirjutatud teegid olid „Arduino“, „Preferences“, „BLEDevice“, „BLE2902“, „Adafruit_GFX“, „Adafruit_NeoMatrix“, „Adafruit_NeoPixel“. „Arduino“ teeki kasutab PlatformIO, et saaks kasutada kõiki Arduino platvormile loodud funktsioone ja muutujaid ilma Arduino IDE'd kasutamata. „Preferences“ teeki kasutatakse sardtarkvaras, et salvestada muutujaid ja nende väärtuseid ESP32 mikrokontrolleri püsimällu. BLE- eesliitega teeki on kasutatud seadme ja mikrokontrolleri vahelise Bluetooth-ühenduse ja andmevahetuse haldamiseks. Adafruit'i teegid on kasutusel LED-tablo juhtimiseks ehk tablool sisu kuvamine, heleduse muutmine ning sisse- ja väljalülitamine.

Mobiilirakenduse poolt kasutatud teeki oli üle 45, kuna React enamasti ainult nendele toetubki. Nii paljude teekide tõttu ei hakata neid siin eraldi välja tooma, kuid need on leitavad loodud GitHub hoidlas lähtekoodi juurkaustast „package.json“ failist.

6 Tööprotsess

Lõputööga samaaegselt kulges ülikoolis projektiaine „Arvutisüsteemide projekt“, seetõttu kasutas autor võimalust ning kombineeris lõputöö ja projekti üheks. See tähendas seda, et edaspidi hakkas arendus toimuma kahenädalaste iteratsioonidena, kus kord kahe nädala jooksul käis autor ülikoolis kohapeal projekti vahepealseid arenguid tutvustamas. Antud arendusvorm sobis autorile hästi, kuna see motiveeris igal nädalal uute tulemuste saavutamiseks midagi tegema ning tehtud tööle ka tagasi vaatama.

Tööprotsess jagunes kolmeks peamiseks osaks: ettevalmistus, arendus ning viimistlus. Arenduse osas on eraldi välja toodud mobiilirakenduse arendus ning sardtarkvara arendus, mis toimusid vaheldumisi. Ajaliselt oli tööprotsessi kõige mahukam osa mobiilirakenduse arendus, mis võttis vähemalt 70% kogu ajast. See võis olla tingitud sellest, et mobiilirakenduse arendusega oli autoril varasemast kõige vähem kogemust, kuid sardtarkvara ning muude tegevustega oldi rohkem kursis.

6.1 Ettevalmistus

Esimese sammuna uuriti, milliseid komponente antud ülesande realiseerimiseks vaja on. Selleks suheldi kaastudengitega ning uuriti internetist sarnaseid projekte. Kui esialgsed komponendid olid välja valitud, telliti need ära ning samal ajal hakati uurima mobiilirakenduse arenduse kohta. Kõigepealt jõudsid kohale esialgsed mikrokontrollerid. Kahjuks selgus aga testimisel, et tegu oli praakidega, mille järel tehti veel uurimist ja telliti uued.

Seejärel saabus esimene tellitud LED-tabloo, mis oli juba valmistoode koos rakendusega, kuid see osutus sobimatuks. Seetõttu telliti projekti jaoks sobivam LED-tabloo ning üks ka varuks, juhuks kui peaks midagi katki olema, või tekkima soov laiendada tablood kõrgemale resolutsioonile. Uued tooted, mis kohale jõudsid, olid töökorras ning võimaldasid arendusega alustada.

Sardtarkvarale ennetavalt struktuuri ei loodud, kuna ei osatud hinnata, milliseid funktsioone programmikood täpselt täitma peaks. Mobiilirakenduse jaoks mõeldi välja

esialgsed kasutusjuhud ja visandati diagramm, millest arendusel juhinduda. Lisaks tehti kokkulepped ülikooliga mobiilirakenduse arendamiseks IT-Kolledži ruumides, kuna autori arvutil vajaminev iOS operatsioonisüsteem puudus. Kogu ettevalmistus koos kõigi komponentide tarnimisega kestis umbes üks kuu.

6.2 Mobiilirakenduse arendus

Mobiilirakenduse arendusel oli mitmeid variante, mille vahel valida: natiivne (inglise keeles *native*) rakendus, platvormist sõltumatu rakendus või progressiivne veebirakendus. Seoses rakendusele seatud nõuetega langeb progressiivse veebirakenduse variant aga kohe välja, kuna iOS veebibrauserid ei toeta Bluetooth'i ja seega ei saaks LED-tablood juhtmevabalt kasutada [31]. Natiivne rakendus luuakse ühele kindlale platvormile selle platvormi arenduseks mõeldud keeltega, milleks iOS puhul on Objective-C ja Swift ning Androidi puhul Java ja Kotlin. Natiivse rakenduse puhul on positiivne see, et see võimaldab kasu saada kõigist seadme enda funktsionaalsustest, mis teeb rakenduse kiireks ja visuaalselt süsteemi sobituvaks. Taoline rakendus seab aga kitsendused tuleviku jaoks, kui peaks soov tekkima mõnele teisele platvormile laieneda. Sel juhul peaks terve koodi uues keeles ümber kirjutama [32]. Lisaks ei olnud autoril natiivse rakenduse arenduseks mõeldud programmeerimiskeeltega varasemat kogemust, seega otsustas ta kolmanda variandi - platvormist sõltumatu rakenduse kasuks.

6.2.1 Tehnoloogia valik

Platvormist sõltumatu mobiilirakenduse arenduseks valiti Meta Platforms'i poolt loodud vabavaraline raamistik React Native. Tegu on ühe populaarseima platvormist sõltumatu raamistikuga, mis võimaldab ühe koodibaasiga arendada rakendusi mitmele erinevale platvormile, nagu iOS, Android, Windows. Antud raamistik võimaldab arendada mobiilirakendusi nagu veebilehti, tuntud veebiraamistiku React'i abil JavaScripti või TypeScripti, HTML'i ja CSS'i kasutades. Raamistik kasutab vahekihti, mis tõlgib veebirakenduse stiilis koodi ümber mobiiliplatvormile arusaadavatesse juhitesse kasutades selleks platvormipõhiseid teke. Lisaks on arendusprotsess kiire, sest saadaval on reaalaajas värskendamine, mis tähendab seda, et kõige uuemate koodimuudatuste nägemiseks ei pea rakendust iga kord uuesti kompileerima [33].

Valikus oli ka teisi platvormist sõltumatuid raamistikke, nagu Google'i Flutter, Microsofti Xamarin ja Ionic. Flutteri kasutamine oleks tähendanud uue

programmeerimiskeele Dart õppimist, mida autoril lõputöö perioodil aega teha ei oleks olnud. Xamarini probleemiks oli see, et selle tugi lõppeb 1. mail 2024 ning see asendatakse .NET MAUI raamistikuga, seega ei oleks selles keeles arendus olnud tuleviku perspektiivis väga mõttekas. Samuti kasutab Xamarin C Sharpi, mis on autorile võõras programmeerimiskeel. Ionic'u puhul sai saatuslikuks vajalike teekide puudus, et arendada Bluetooth LE toega mobiilirakendus [34].

Kuigi autoril varasem kogemus React'iga puutus, oli olemas kogemus veebirakenduse loomisest tudengiprojektis, kus kasutati samuti JavaScripti, kuid teise populaarse veebiraamistiku, Vue.js abil. Vue.js ja React raamistikel on nii mõndagi ühist, seega ei pidanud autor arenduses täiesti nullist alustama [35]. Siiski kulusid esimesed nädalad arendusest uue raamistiku tundma õppimiseks, sest ka mobiilirakenduse arendus kui selline oli autorile uus. Kuigi React Native võimaldab arendada platvormist sõltumatult, arendati antud rakendust aja kokkuhoiu huvides spetsiifiliselt iOS operatsioonisüsteemile, seega on võimalik, et Androidi või mõne muu operatsioonisüsteemiga rakenduse kasutamiseks on vaja programmikoodi pisut täiendada.

Rakenduse arendusega kiirelt pihta hakkamiseks soovitati React Native'i kasutusjuhendis kasutada Expo, mis on React Native'i ümber ehitatud tööriistade ja teenuste komplekt. Expo abil on paari käsuga võimalik esialgne rakenduse mall simulaatoris avada [36]. Seda autor ka tegi ning selle kasutusmugavuse tõttu kasutati Expo terve ülejäänud arendusperioodi vältel.

6.2.2 Kujunduse valik

Mobiilirakenduse värvilahenduseks valiti tume teema koos erinevate siniste toonidega. Taustadena kasutatakse sujuvaid värviüleminekuid (inglise keeles *gradient*). Tume teema tagab selle, et loodud visuaalid paistavad hästi välja ning pimedal ajal, mis on peamine eeldatav LED-tabloode kasutusaeg, on mugavam ka silmale.

Kujunduste loomisel lähtus autor enda maitsest, hetketrendidest ning üldistest UX põhimõtetest [37]. Järgitud on ühtset vormindust, sarnast komponentide joendamist, kasutatud ühtset värvigammat ning piiratud ühe kirjagarnituuri erinevate paksusastmetega. Üldiselt jälgib lõpprakendus makettidel loodud stiili, kuid protsessi käigus kohandati ja mugandati teatud kuvasid vastavalt vajadusele. Rakenduse kuvadel

ja komponentides kasutatavad ikoonid on võetud expo/vector-icons vabalt kasutatavate ikoonide teegist [38]. Kasutatavaks kirjagarnituuriks on „Inter“, mida pakub Google ning mis on vabavaraline [39]. Kirjagarnituur valiti seetõttu, et see on nii veebi- kui ka mobiilirakendustes laialt levinud ning näeb kaasaegne välja. Lõputöö käigus ei mõeldud rakendusele välja lõplikku nime ega logo, kuna arendusfaasis ei ole plaanis seda levitama hakata. Nimesid „LEDMASTER“ ning „BTLED“ võib pidada esialgseteks kohahoidjateks, kuni tegelik nimi selgub.

6.2.3 Protsess

Esiteks seati üles arenduskeskkond. Vajalik tarkvara ehk VSCode ja Xcode olid ülikooli arvutis juba olemas, seega tuli ainult installeerida vajalikud laiendused ning võtta kasutusele Expo poolt pakutud mall arenduse alustamiseks. Arenduskeskkond tuli seadistada nii, et saaks kasutada nii Xcode'i seadme simulaatorit, mis võimaldab arvutimonitoril näha ära kiiremad muutused, kui ka reaalselt nutiseadet, sest simulaatoril puudub Bluetooth-ühenduse võimekus. Antud seadistuse korraldamine oli arenduse alustamise kõige probleemsem osa. Et ühendus lõpuks toimima saada, pidid seadmed olema samas WiFi võrgus, mis pidi olema piisavalt avalik, et ühes võrgus olevad seadmed üksteist näeksid, kuid piisavalt privaatne, et saaks seda turvaliselt kasutada. Lisaks sellele pidid nutiseade ja arvuti olema ühendatud juhtme abil.

Peale seda alustati baasfunktsionaalsuse loomisega ehk seadmega Bluetooth-ühenduse saavutamine ning hoidmine. Selleks loodi ka avaleht, kus oleks võimalik seadet ühendada. Alguses olid suureks abiks YouTube'i keskkonnas kanali „Dan's React Native Lab“ videod, kus loodi näidisprojektid täpselt React Native raamistikus Bluetooth Low Energy't kasutades [40]. Nende põhjal sai autor esialgse arusaama Bluetooth Low Energy toimimisest ning juhtmevaba ühenduse mikrokontrolleriga tööle. Ühenduse saavutamiseks ja kõigi Bluetooth-funktsionaalsuste jaoks mobiilirakenduses kasutati React-Native'ile loodud teeki react-native-ble-plx [41].

Nõuetest tingituna implementeeriti kohe arenduse alguses rakenduse mitmekeelsus. Selleks võeti kasutusele paljudes JavaScripti raamistikutes leitav i18next lokalisatsiooni teek. Teegi jaoks tuli luua ka eesti- ja ingliskeelne keelefail, mis sisaldab kõiki erinevaid rakenduse teateid, tekste sõnumeid ehk kõike tõlgitavat. Teek võimaldab mõne lihtsa funktsiooniga muuta kogu rakenduse keelt, kui sobivad keelefailid on olemas [42].

Järgmisena lõi autor erinevad kuvad eelnevalt loodud kavandite järgi ning seejärel hakkas kuvadesse funktsionaalsusi lisama. Selleks pidi looma funktsioonid, mis vastutavad andmete saatmise eest mobiilirakendusest mikrokontrollerini. Lisaks tuli kuvadele luua erinevad graafilised komponendid, nagu nupud, lõuend kujunduse loomiseks, kujunduse eelvaadete komponendid ning loendi komponendid. Kogu arendusprotsessi vältel võttis kõige rohkem aega just taoliste graafiliste komponentide loomine, kus on ka mingi puutefunktsionaalsus. Hilisema arenduse käigus tehti mõned kuvad ümber modaalakendeks (rakenduse seaded, keelevalik), kuna leiti, et need ei vajanud nii suurt ekraanipinda. Lisaks sai autorile arenduse käigus ka tuttavaks React'i eripära – enamik eelloodud komponente või teeke, mida oleks vaja oma komponentide baasiks kasutada, tuleb internetist otsida ning rakendusse sisse importida. Ka see tegevus võttis arendusprotsessi käigus omajagu ajast.

Ühel hetkel selgus, et lokaalsed muutujad komponentides ei ole hea lahendus ning oleks vaja mingit ühtset globaalsete muutujate süsteemi. Tuli kasutusele võtta ladu (inglise keeles *store*), mis võimaldab hoiustada rakenduse olekut (globaalseid muutujaid) ning sellega seotud funktsioone ühes kohas, võimaldades sellele ligipääsu igast rakenduse komponendist. React Native raamistikus pakub seda võimalust teek nimega Redux [43]. Lao ülesseadmiseks tuli luua eraldiseisev failide komplekt, mis kirjeldas ära kõik muutujad, mida soovitakse globaalsetena hoida ning ka funktsioonid, mida nende muutujatega võimalik kasutada on. Rakenduse iga osa jaoks loodi vastav viil – niimoodi kutsutakse üht osa globaalsest olekust, kuhu lähevad selle osa poolt kasutatavad muutujad. Näiteks on rakenduses Bluetooth'i viil, mis hoiab olekut ühenduse staatuse, Bluetooth'i oleku ning ühendatud seadmete kohta.

Rakendus vajab ka funktsionaalsust, et säilitada juba loodud kujundused ning muu kasutaja poolt sisestatud info nutiseadme püsivus. Selleks tuli kasutusele võtta teek Redux Persist, mis võtab eelnevalt lattu salvestatud globaalse oleku ning hoiustab seda nutiseadme mälu, et pärast rakenduse sulgemist ja uuesti avamist ei vaataks vastu tühi leht [44].

Mobiilirakenduse arenduse käigus toimus jooksvalt ka selle toimimise testimine LED-tablooga. Tihtipeale tulid sellisel moel välja erinevad vead, mille algpõhjuseid tuli otsida

nii mobiilirakendusest kui ka sardtarkvarast. Vigade otsimisel mobiilirakenduses oli abiks JavaScripti käsk „`console.log()`“, mille saab lisada koodiridade vahele, et väljastada mõne muutuja väärtus, või näha, kuhu maale kood jõuab enne vea tekkimist. Käsu täitmisel väljastatakse VSCode'i või Xcode'i konsooliaknasse käsu sulgude sees olev väärtus. Mobiilirakenduse funktsioonidele ja üldisele toimimisele aja kokkuhoiu huvides teste ei kirjutatud, kuid hilisema arenduse käigus on neid võimalik lisada.

6.3 Sardtarkvara arendus

Käesoleva lõputöö raames mikrokontrolleri sardtarkvara arendusel lähtuti koodi lihtsusest, mitte jõudlusest. See tähendab seda, et võeti kasutusele võimalikult palju abstraktsioone ning standardsetele C-keele andmetüüpidele eelistati C++ objekte. Lihtsuse põhimõttest lähtuti seetõttu, et arendus oleks võimalikult kiire. Lisaks eeldati, et kui kõik asjad toimima saab, siis on edasiarenduse käigus võimalik alati jõudlust parandada.

Sardtarkvara arendust alustati mikrokontrollerile erinevate testprogrammide kirjutamisest. Esmalt testiti mikrokontrolleri enda funktsionaalsusi: Bluetooth'i, WiFi toimimine, sisseehitatud LED-i vilgutamine ja värvi muutmine, väljaviikude toimimine. Seejärel prooviti erinevaid funktsioone Adafruit'i teekidest LED-tablool sõnumi või visuaali kujutamiseks.

Veendutud, et kõik riistvaralised komponendid toimivad, liiguti edasi Bluetooth-ühenduse implementeerimise juurde. Esialgu võeti kasutusele Arduino poolt pakutud näidiskood, kuid arenduse käigus muudeti seda vastavalt vajadusele. Kui ühendus toimis, lisati funktsioon mikrokontrolleri sisseehitatud RGB (*Red Green Blue*) LED juhtimiseks, et ühenduse staatust näidata. Seejärel alustati LED-tablool sisu kuvamise funktsioonide kirjutamisega. Enamik nendest funktsioonidest olid algsel kujul leitavad Adafruit'i poolt pakutavatest teekidest, kuid neid tuli muuta, et need toimiksid korrektselt Bluetooth'i kaudu vastuvõetava informatsiooniga. Alustati lihtsamate funktsioonidega, nagu teksti kuvamine tablool, teksti värvi muutmine, tablo heleduse muutmine ja sisseväljalülitamine. Seejärel liiguti tablool visuaali, liikuva teksti ning animatsioonide kuvamiseni.

LED-tablool sisu kuvamise funktsioonid esialgsel kujul toimima saadud, kirjutati funktsioonid mikrokontrolleri püsिमälus info hoiustamiseks. Hoiustatavaks infoks on viimasena kuvatud animatsioon, visuaal või tekst ning arvuline väärtus, mis näitab, kas viimasena kuvati animatsiooni, visuaali või teksti. Antud funktsionaalsus lisati seetõttu, et pärast LED-tablool väljalülitamist või volukatkestust saaks musta pildi asemel kuvada varasemalt tablool olnud infot. Lisaks plaanitakse lõputöö edasiarenduse käigus kasutada sarnaseid püsिमälu muutmise funktsioone, et hoida mikrokontrolleris ka infot viimati ühendatud LED-tablool tüüpi kohta, mida saaks sisestada mobiilirakenduses.

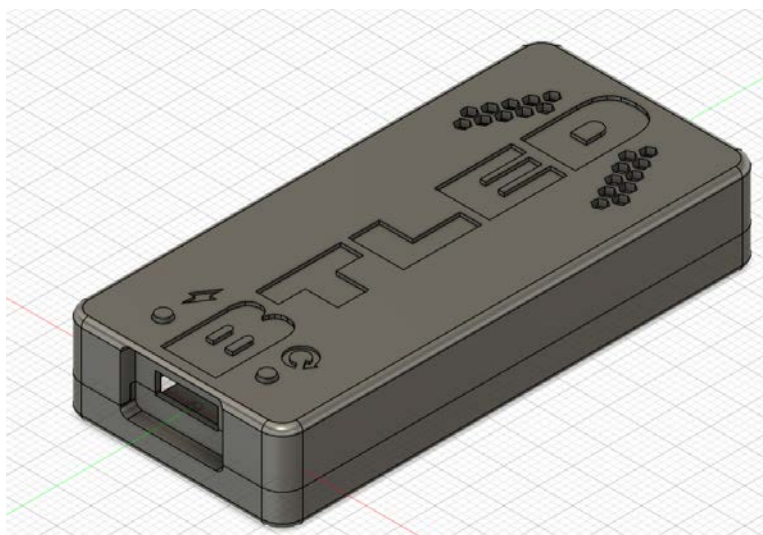
Suurimad probleemid sardtarkvara arendusel esinesid kahe tuumaga töötamisel. ESP32-S3 mikrokontrolleril on kaks tuuma: tuum 0 ja tuum 1. Eraldi ülesandeid (inglise keeles *task*) erinevatel tuumadel programmiosade jooksumiseks esialgu ei loodud, kuna käsu „xPortGetCoreID()“ abil oli näha, et Arduino oma funktsioonid, nagu „loop()“ ja „setup()“ jooksevad tuumal 1. Bluetooth-funktsionaalsus jooksis aga tuumal 0. Probleem seisnes selles, et mobiilirakendusest tulevad andmed võtsid palju mälu ning kuna funktsioon, mis tegeles andmetöötlusega ja tablool sisu kuvamisega, kutsuti välja otse Bluetooth'i andmeid vastuvõtva funktsiooni seest, siis sai mälu otsa, sest Bluetooth'i funktsioonidele ei olnud nii palju mälu kuhjast eraldatud. See ja sarnased probleemid lahendati lisades väljakutsutavatele funktsioonidele vastavad globaalsed tõeväärtuse andmetüübiga muutujad, mis vaid tõese väärtuse korral funktsiooni käivitavad ning siis väärtuse uuesti vääraks muudavad. Selle asemel, et Bluetooth'i funktsioonist teisi funktsioone välja kutsuda, muudetakse seal lihtsalt globaalse muutuja väärtust, mille järel kutsutakse funktsioon välja hoopis „loop“ tsüklist, mis jookseb teisel tuumal.

Vahepeal tekkisid konfliktid ka mikrokontrolleri sisseehitatud RGB LED-i ning LED-tablool vahel, kuna mingil põhjusel saadeti LED-tablool kuvatav info signaal ka väljaviigule, mis sisseehitatud LED-i juhib. Selle probleemi lahendamiseks kasutati esialgse ESP32 teegi poolt pakutud funktsiooni „neopixelWrite“ asemel Adafruit'i teegi „Adafruit_NeoPixel“ funktsiooni „pixel.write“ ning enne seda tehti sama teegi abil ka sisseehitatud LED-ile algseadistus, kus määrati õige väljaviik. Peale seda taolised konfliktid lõppesid.

6.4 Viimistlus

Mobiilirakenduse puhul toimus viimistlus tegelikult kogu arenduse käigus, kuna komponentide lisamisel tuli tihtipeale eelmisi komponente muuta, et need paremini kokku sobituks. Kuna mobiilirakendus jääb veel edasi arendusfaasi, siis lõppviimistlust sellele ei tehtud. Sardtarkvara puhul jagati mitmed funktsioonid laiali eraldiseisvatesse klassidesse, parandati koodi struktuuri, lisati päsefailid ning loodi GitHub'i hoidlasse programmi kirjeldav dokument.

Arendusplaadi viimistluseks loodi sellele Fusion360 modelleerimistarkvara ja 3D-printeri abil korpus. Autor pidas seda vajalikuks, et varjata ära liigne elektroonika ning luua prototüübist parem üldmulje (vt Joonis 3).



Joonis 3. Fusion360 modelleerimistarkvaras loodud arendusplaadi korpus.

Korpuse loomisel kasutati veebimaterjalide abi ning autori isiklikke kujunduseelistusi. Kuna autor ei olnud varem midagi sellist teinud ning antud arendusplaadile eelloodud korpust internetis ei leidunud, ei tulnud esimesel printimisel kõik ideaalselt välja. Arendusplaat ei olnud korpuse sees korralikult fikseeritud ning korpusele ei loodud ka mingit mehhanismi, et ülemine ja alumine omavahel koos püsiksid. Lisaks oli korpuse pealmine kiht liiga õhuke ning selle peal olevad sümbolid liiga väikesed. Selle parandamiseks modelleeris autor ka korpuse teise versiooni, kuid käesoleval hetkel ei ole seda veel jõutud printimisse saata (vt Joonis 4).



Joonis 4. Fusion360 modelleerimistarkvaras loodud arendusplaadi korpus vol. 2.

Viimistluse käigus muudeti ka ühendusi LED-tabloole ning arendusplaadi vahel. Kui varem sai LED-tablo lisaks andmetele ka voolu arendusplaadilt, mis testimise jaoks oli sobilik, siis suurema voolutarbe võimaldamiseks ning mikrokontrolleri kahjustamise vältimiseks ühendati LED-tablo eraldi toitekaabliga kasutades ära tablo keskel paiknevaid ühenduskohti. Enne vooluallikaga ühendamist lisati paralleelselt ka 1000-mikrofaradine kondensaator, et leevendada äkilisi voolumuudatusi, mis võivad LED-tabloole kahju teha. Lisaks lisati arendusplaadi ning LED-tablo ühenduse vahele 330-oomine takisti, mis peaks vähendama võimalikke mürasid signaalis. Kondensaatori ja takisti lisamine on soovitatud Adafruit'i poolt NeoPixel tüüpi LED-ide üles seadmisel, millest ka antud LED-tablo koosneb [45].

LED-tablo välimuse viimistluseni antud lõputöö käigus ei jõutud ning see jääb võimalikesse edasiarendustesse. Viimistlusena on plaanis lisada sellele mugavamaks käsitlemiseks ümbris ning LED-ide ette lisakiht, et valgust pisut hajutada.

7 Sardtarkvara

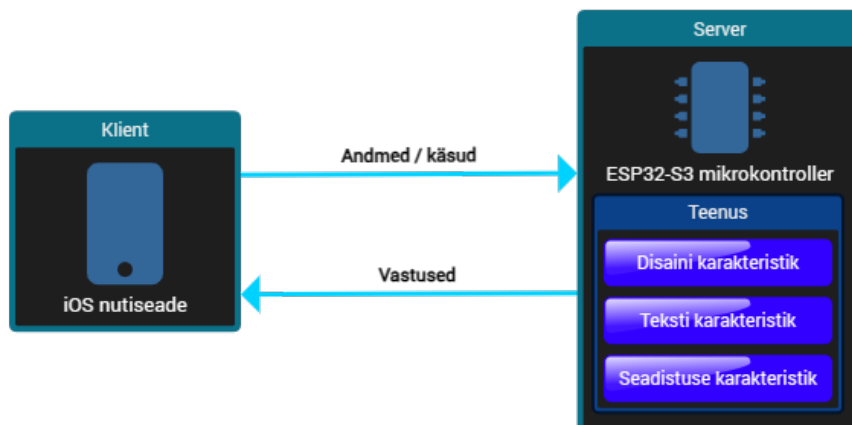
Sardtarkvara jookseb ESP32 mikrokontrolleris ning on kirjutatud Arduino platvormile C++ keeles, kasutades selleks VSCode'i ning PlatformIO laiendust. Selle eesmärk on Bluetooth LE-d kasutades vahendada infot mobiilirakenduse ning LED-tabloo vahel. Järgnevates alapeatükkides on kirjeldatud ühenduse toimimist, välja toodud programmi juhtloogika ja loodud funktsioonid ning nende otstarve.

7.1 Bluetooth-ühendus

Mikrokontrolleri ning nutiseadme vahel kasutatakse andmevahetuseks Bluetooth LE ühendust. Antud ühenduse tüüp valiti kahel põhjusel: esiteks spetsifitseeriti seda nõuetes ning teiseks oli see ainus Bluetooth-ühenduse tüüp, mida soetatud mikrokontroller ESP32-S3 võimaldas [46].

Bluetooth LE protokoll koosneb paljudest kihtidest ning sisaldab endas ka pakettide kontrollimist, mis tähendab, et seda ei pea ise rakenduses andmete vastuvõtmisel implementeerima. Bluetooth LE võimaldab andmeedastuskiirust kuni 1.4 Mb/s ehk visuaalide ning animatsioonide saatmisega erilise viivitusega ei tohiks probleemi olla [47].

Oluline kiht, mis kirjeldab andmeedastust on GATT (*Generic Attribute Profile*). Selle kihi puhul on defineeritud kaks seadet: klient ja server. Klient edastab tavaliselt mingeid andmeid ning server muudab nende andmete põhjal oma olekut. Loodava rakenduse puhul on kliendiks nutiseade ning serveriks mikrokontroller (vt Joonis 5).



Joonis 5. Diagramm Bluetooth LE GATT kihi kasutusest loodud süsteemis.

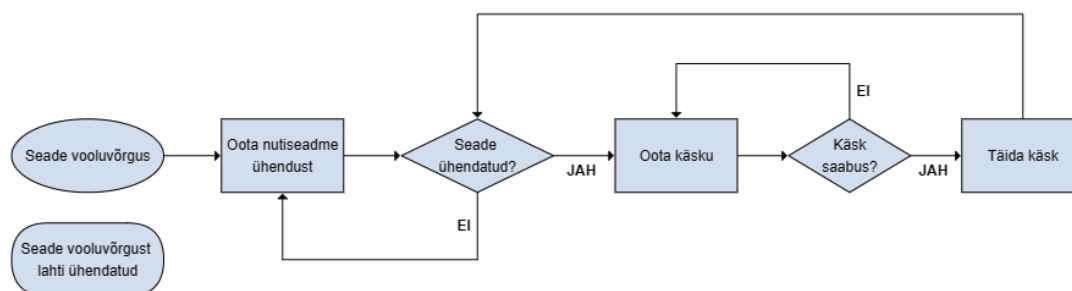
Andmevahetus toimub nii, et server omab profiili, millele luuakse unikaalse identifikaatoriga teenus ning antud teenuse külge lisatakse karakteristikuid erinevate andmete jaoks. Igal karakteristikul on samuti unikaalne identifikaator, kuid on ka omadused, mis näitavad ära, milliseks tegevuseks (lugemine, kirjutamine, märguanne) on antud karakteristik mõeldud. Karakteristikule võib omakorda lisada ka kirjeldaja (inglise keeles *descriptor*), mis seab täiendavad täpsustused. Kui serveri poolt on Bluetooth-teenus käivitatud, hakkab server ennast reklaamima. Kui klient serveri leiab ning sellega ühendab saab ta teenuse ning karakteristiku unikaalset identifikaatorit teades serverile andmeid saata ning vastavalt karakteristikule teab server, mida kättesaadud andmetega teha [48].

Loodud rakenduses kasutatakse disaini karakteristikut, et mikrokontroller saaks vastu võtta animatsioone ja visuaale, teksti karakteristikut, et vastu võtta sõnumeid, mida kuvada ning seadistuse karakteristikut, et vastu võtta LED-tabloo ereduse väärtust või sisse-/väljalülituse käsku.

7.2 Juhtloogika

Loodud sardtarkvara juhtloogika koosneb kolmest peamisest tsüklist: ühenduse ootamine, käsu ootamine ning ühenduse kontrollimine (vt Joonis 6). Esimene tsükkel käivitub siis, kui LED-tabloo ja mikrokontroller on vooluvõrku ühendatud ning alglaadimine lõpetatud. Tsükli kontrollitakse, kas nutiseade on vahepeal ühendanud mikrokontrolleriga. Ühenduse korral käivitub teine tsükkel, kus oodatakse käsku

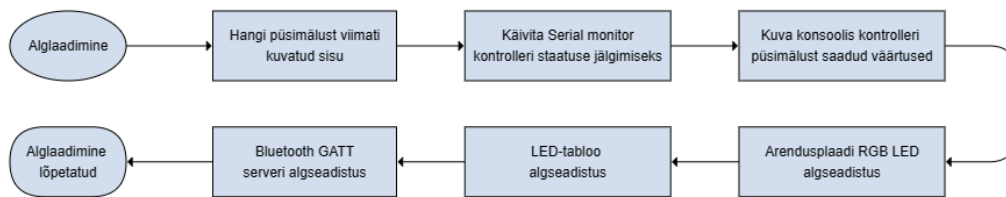
nutiseadmelt. Käsk võib olla näiteks LED-tablool sisu kuvamine või LED-tablooleku või heleduse muutmine, kuid ka ühenduse katkestamine mobiilirakendusest. Pärast käsu saabumist see täidetakse, näiteks kuvatakse LED-tablool sisu või ühendatakse seade lahti. Seejärel kontrollitakse uuesti, kas seade on ikka ühendatud. Kui seade on ühendatud, korratakse sama tsüklit, kui mitte, siis minnakse tagasi esimese tsükli juurde, mis tähendab ühenduse ootamist. See on väga üldine kontseptsioon ning detailsema ülevaate saamiseks on loodud spetsiifilisem juhtloogika voodiagrammi Lisas 2.



Joonis 6. Sardarkvara juhtloogika üldistatud voodiagramm.

Algladimine

LED-tabloole vooluvõrku ühendamisel toimub esimese sammuna algladimine (vt Joonis 7). Algladimise käigus hangitakse esiteks mikrokontrolleri püsivõrgu LED-tabloole viimati kuvatud sisu ja salvestatakse muutujatesse. Kui viimati kuvatud sisu puudub, saavad täisarvu andmetüübiga muutujad väärtuseks null, *String* tüüpi muutujad tühja *String*'i ning tõeväärtuse andmetüübiga muutujad väärtuseks „väär“. Seejärel käivitatakse jadapordi monitor, mille peamine otstarve on silumine, ning kuvatakse konsolis püsivõrgu saadud väärtused. Siis algseadistatakse arendusplaadi sisseehitatud RGB LED, et see saaks Bluetooth ühenduse staatuse muutumisest märku anda. Järgmisena algseadistatakse LED-tabloole, mille käigus määratakse heledus, kirjagarnituur ning pannakse LED-tabloole kuvama viimati kuvatud visuaali, teksti või animatsiooni, kui see eksisteerib. Vastasel juhul ilmub tabloole tekst „On“. Viimaseks algladimise etapiks on Bluetooth serveri algseadistus, kus luuakse vajalikud teenused ja karakteristikud ning alustatakse seadme reklaamimist, et seda oleks võimalik nutiseadme poolt leida.



Joonis 7. Sardtarkvara algladimise lihtsustatud voodiagramm.

7.3 Klassid ja nende funktsioonid

Sardtarkvaras loodi kuus klassi: „MemoryFunctions“, „MyServerCallbacks“, „MyDesignCallbacks“, „MyTextCallbacks“, „MySetupCallbacks“ ja „MyAnimations“.

Klass „MemoryFunctions“ hoiab endas funktsioone, mis hõlmavad mikrokontrolleri püsimaluga operatsioonide tegemist. Neid funktsioone on kahte tüüpi: andmete salvestamine püsimalu ning andmete hankimine püsimalust. Antud klass kasutab püsimalu operatsioonide jaoks „Preferences.h“ teeki. Iga funktsiooni lõpus ning alguses on käsud mälu piirkonna avamiseks ning sulgemiseks ning nende vahele jääb kas lugemine või kirjutamine sellesse piirkonda.

Klass „MyServerCallbacks“ hoiab endas kahte funktsiooni, mis on pärit „BLEDevice“ teegist ning kutsutakse välja teatud sündmuste puhul. Antud juhul on nendeks funktsioonideks „onDisconnect“ ja „onConnect“, mis kutsutakse välja vastavalt nutiseadme lahtiühendamisel mikrokontrollerist ja nutiseadme ühendamisel mikrokontrolleriga. Funktsioonide parameetrik on viit sardtarkvara algladimisel loodud GATT-serveri profiilile ning selle abil on võimalik kindlaks teha, millal ühendamise sündmus toimub. Funktsioonide sisu on muuta globaalset tõeväärtuse tüübiga muutujat „deviceConnected“, mille põhjal otsustatakse sardtarkvara põhitsükklis, kas jätkata ühenduse ootamist, või kutsuda välja mõni funktsioon.

Klassid „MyDesignCallbacks“, „MyTextCallbacks“, „MySetupCallbacks“ sisaldavad kõik endas ühtainust funktsiooni „onWrite“, millele antakse parameetrik viit GATT-serveri karakteristikule. See funktsioon kutsutakse välja, kui parameetrik olevale karakteristikule saadetakse nutiseadmest Bluetooth'i kasutades andmeid. Klasside „MyDesignCallbacks“ ja „MyTextCallbacks“ „onWrite“ funktsioon kutsub välja funktsiooni LED-tablool kujunduste või sõnumite kuvamiseks ning klassi

„MySetupCallbacks“ puhul kutsutakse välja funktsioon LED-tabloo oleku või ereduse muutmiseks.

Klass „MyAnimations“ on mõeldud animatsioonide loomise funktsioonide hoiustamiseks. Hetkel sisaldab antud klass vaid ühte funktsiooni, milleks on „policeCon“. Antud funktsiooni nimi tuleneb sellest, et vilgutatakse kordamööda sinist ja punast LED'i. Funktsiooni kasutatakse mikrokontrolleri sisseehitatud LED'i vilgutamiseks, kui nutiseade ei ole mikrokontrolleriga ühendatud. See annab kasutajale märku, et on võimalik seadmega ühendada, või ühendus on katkenud. Klassis on ainult üks funktsioon, kuna lõputöö käigus ei jõutud arendusega LED-tablool kuvamiseks mõeldud erinevate animatsioonide funktsioonide loomiseni.

7.4 Eraldiseisvad funktsioonid

Kokku loodi sardtarkvaras lisaks klassifunktsioonidele 13 eraldiseisvat funktsiooni. Edasise arenduse käigus proovitakse ka need funktsioonid klassidesse ära jaotada, et rakendus oleks paremini hallatav, kuid antud lõputöö raames seda teha ei jõutud. Järgnevalt tutvustatakse olulisemaid loodud funktsioone täpsemalt.

Funktsioon `initBLE`

Antud funktsiooni eesmärk on käivitada ESP32 mikrokontrolleris GATT-server, mis võimaldaks juhtmevaba ühenduse ja andmevahetuse nutiseadme ning LED-tabloo vahel. Funktsioon baseerub suuresti „BLEDevice“ teegil ning näidiskoodil. Funktsioonis luuakse server, üks unikaalse identifikaatoriga teenus ning sellele teenusele kolm erinevat LED-tablooga seotud karakteristikut, millest üks on kujunduste jaoks, üks sõnumite jaoks ning üks seadistuse jaoks. Peale seda käivitatakse loodud teenus ning alustatakse teenuse reklaamimist, et seda oleks võimalik mobiilirakenduse kaudu leida [49].

Funktsioon `turnMatrixOn`

Antud funktsiooni eesmärk on LED-tabloo sisse lülitada ehk taastada olek, mis oli enne LED-tabloo väljalülitamist. Väljalülitamisena läheb arvesse nii mobiilirakenduse kaudu tabloo väljalülitamine kui ka voolust lahti ühendamine. Funktsioon kontrollib esmalt muutujat „matrixOn“, mis näitab kas LED-tabloo on juba sisselülitatud. Kui on, siis väljutakse funktsioonist. Vastasel juhul salvestatakse globaalsesse muutujasse „currentDisplayType“ viimati LED-tablool kuvatud sisu tüüp (kujundus, tekst), mis

võetakse mikrokontrolleri püsimalust. Seejärel vaadatakse tüübi järgi, kas viimati kuvati kujundust või teksti ning selle põhjal kutsutakse välja kas LED-tablool kujunduse või teksti kuvamise funktsioon. Kui tüüp on tundmatu, kuvatakse LED-tablool valge värviga sõnum „On“, mis näitab, et tablooo töötab. Seejärel muudetakse muutuja „matrixOn“ väärtus tõseks, saadetakse konsooli sõnum „Turning on...“ ning lahkutakse funktsioonist.

Funktsioon turnMatrixOff

Funktsiooni eesmärk on LED-tabloos kõik pikslid kustutada (määrata neile must värv), et tabloost ei tuleks valgust. Esmalt kontrollib funktsioon muutuja „matrixOn“ väärtust ning kui see on väär, siis väljutakse funktsioonist, sest tablooo on juba välja lülitatud. Tõese väärtuse korral seatakse LED-tablooo pikslid mustaks, kasutades teegi „Adafruit_NeoMatrix“ funktsiooni „clear“, värskendatakse tablooo pilti kasutades sama teegi „show“ funktsiooni ning seatakse muutuja „matrixOn“ vääraks. Pärast seda saadetakse ka konsooli sõnum, et LED-tablooo on välja lülitatud ning väljutakse funktsioonist.

Funktsioon setMatrixDesign

Antud funktsiooni eesmärk on kuvada LED-tablool Bluetooth-ühenduse kaudu saabunud kujundus. Funktsiooni parameetrik on *String*-andmetüüpi muutuja, mis sisaldab endas kujunduse värviinfot. Esiteks salvestab funktsioon parameetrik saadud muutuja mikrokontrolleri püsimalu viimati LED-tablool kuvatud kujundusena. Seejärel salvestatakse mikrokontrolleri püsimalu viimati kuvatud sisu tüübiks „Design“. Viimaseks seatakse muutuja „matrixOn“ tõseks, sest LED-tablooo võib olla kujunduse kättesaamisel välja lülitatud, ning kutsutakse saadud kujunduse parameetriga välja funktsioon „stringToBitmap“, mis teisendab parameetri sobivaks andmetüübiks ning kuvab kujunduse LED-tablool.

Funktsioon stringToBitmap

Antud funktsiooni eesmärk on võtta parameetrina saadud *String* andmetüüpi muutuja, mis sisaldab Bluetooth'i kaudu saabunud kujundust, teisendada see sobivaks ning kuvada LED-tablool. Funktsioonis luuakse esiteks sobiva suurusega massiiv kujunduse hoiustamiseks. Seejärel käiakse tsükliga parameetrina saadud muutuja läbi ning salvestatakse iga värvi HEX-kood eraldi kohale massiivis. Pärast seda kasutatakse teegi „Adafruit_NeoMatrix“ funktsioone „drawRGBBitmap“ ja „show“ et kuvada antud

massiivis sisalduvad värviaandmed LED-tablool. Lõpuks vabastatakse ka mälu, mis esialgu uuele massiivile eraldati.

Funktsioon `setMatrixText`

Antud funktsiooni eesmärk on kuvada LED-tablool funktsioonile parameetrina antud sõnum. Esialgu salvestatakse sõnum ning viimati kuvatud LED-tabloole sisu tüübina „Text“ mikrokontrolleri püsimeemoris. Seejärel määratakse parameetri esimesest kuuest sümbolist saadud sõnumi värv, kasutades selleks „Adafruit_NeoMatrix“ teegi funktsiooni „setTextColor“. Järgmiseks seatakse teksti algkoordinaat, et tekst ilmuks tablool õiges kohas ja pärast seda tühjendatakse ekraan eelmisest infost ning kuvatakse eelmainitud teegi „print“ funktsiooni abil sõnum LED-tablool. Lõpuks muudetakse muutuja „matrixOn“ tõeseks, et vältida olukorda, kus tekst võetakse vastu, aga seda ei kuvata, kuna tabloole on välja lülitatud olekus.

Funktsioon `playAnimation`

Antud funktsiooni eesmärk on LED-tablool animatsioon kuvada. Funktsioonis seatakse muutuja „matrixOn“ väärtus tõeseks ning käiakse tsükli abil läbi massiiv, mis sisaldab animatsiooni kaadreid. Iga kaadri puhul kutsutakse välja funktsioon „stringToBitmap“, mis kuvab selle kaadri LED-tablool ning seega tekibki liikuv pilt. Kaadrite vahel kasutatakse ka viidet, mis on võimalik programmi globaalsete muutujate abil määrata.

Funktsioon `setScrollingText`

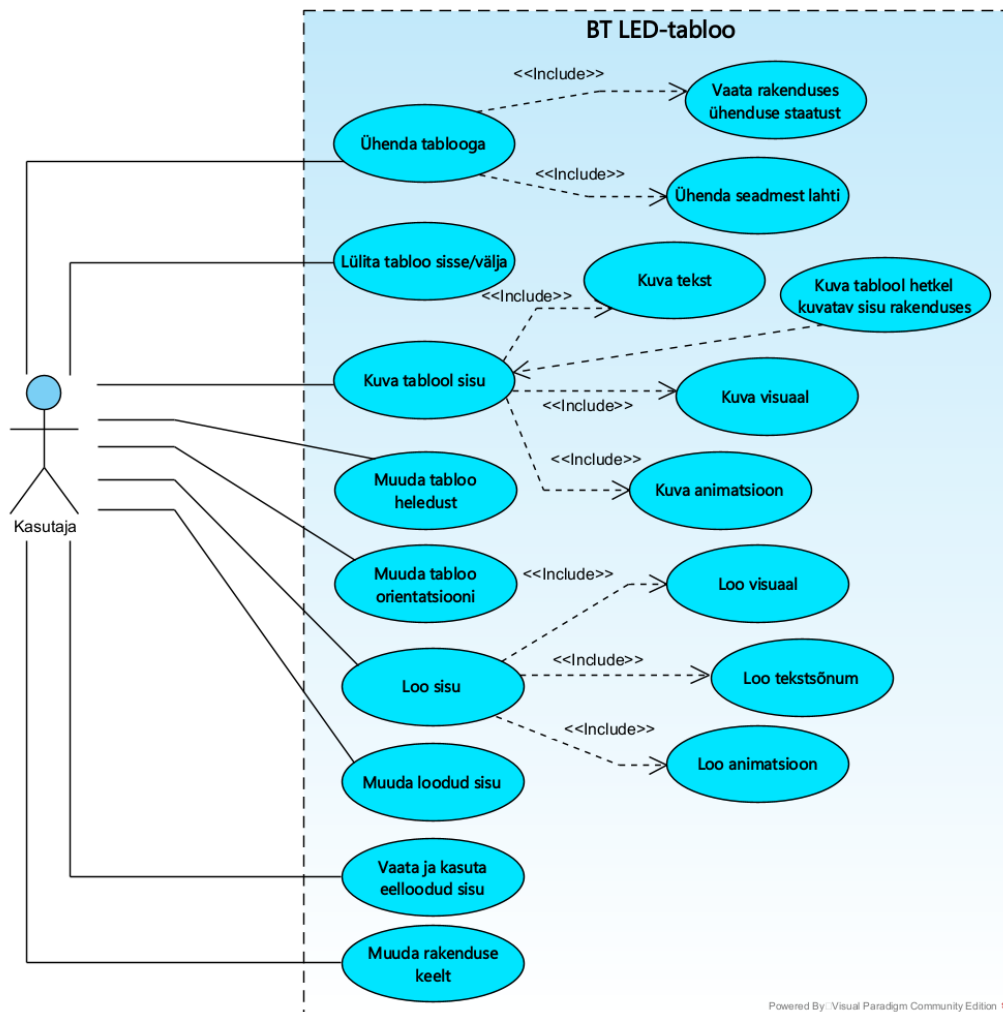
Antud funktsiooni eesmärk on kuvada LED-tablool liikutavat teksti. See funktsioon kutsutakse välja siis, kui sõnum on liiga pikk, et mahtuda ära LED-tabloole. Liiga pika sõnumi puhul liigub sõnum mööda LED-tablood vasakule seni, kuni see on lõppenud ning seejärel alustatakse algusest. Esiteks seatakse funktsioonis muutuja „matrixOn“ väärtus tõeseks, et tulemust näha oleks. Seejärel määratakse sõnumile värv ning arvutatakse sõnumi pikkus pikslites. Kuna ühe tabloole kuvatava tähemärgi laius on enamasti 6 pikslit, siis on sõnumi pikkus pikslites kuuekordne sümbolite arv sõnumis. Pärast seda käivitub tsükkel, kus igas määratakse uus algkoordinaat, kuvatakse sõnum LED-tablool ning vähendatakse algkoordinaati. Sellega tekib kasutajale efekt, et tekst liigub LED-tablool paremalt vasakule.

8 Mobiilirakendus

Käesolev peatükk kirjeldab lõputöö käigus arendatud mobiilirakendust, mida kasutatakse LED-tablool kuvatava pildi muutmiseks. Peatükk koosneb viiest alapeatükist, kus kirjeldatakse mobiilirakenduse kasutusjuhte, arenduseks kasutatud tehnoloogiat, rakenduse arhitektuuri ja komponente ning kuvasid.

8.1 Kasutusjuhud

Käesolevas alapeatükis tutvustatakse mobiilirakenduse võimalikke kasutusjuhte. Paremaks ülevaateks on loodud ka kasutusjuhtude diagramm (vt Joonis 8).



Joonis 8. Mobiilirakenduse kasutusjuhtude diagramm.

Mobiilirakendusel on üks kasutaja roll ning peamisi kasutusjuhte leiti üheksa, millest kolmel on omakorda alamkasutusjuhud. Järgnevalt kirjeldatakse täpsemalt iga kasutusjuhtu.

Kasutusjuht 1: Ühenda tablooga

Kasutaja saab mobiilirakenduses ühendada nutiseadme Bluetooth'i abil LED-tablooga. Selleks peab ta rakenduse avamisel vajutama „Otsi seadet“ nupule. Kui nutiseadmes on Bluetooth sisse lülitamata, avaneb vastava sõnumiga hoiatav aken, kus kasutaja saab vajutada „Seaded“, mille järel suunatakse ta telefoni seadetesse, kus ta saab Bluetooth'i sisse lülitada. Kui Bluetooth on sisse lülitatud, avaneb kasutajale nimekiri saadaolevatest seadmetest. Seejärel peab kasutaja vajutama sobiva seadme kõrval nupule „Ühenda“, et seadmega ühendus luua.

Kui seade on ühendatud, ilmub ühendatud seadme alla seadmete loendis ka vastav teave „Ühendatud“. Ühendatud seadme kõrvale tekib loendis ka nupp „Ühenda lahti“, mis võimaldab kasutajal seadme lahti ühendada. Kui kasutaja ei asu rakenduses enam avalehel, siis saab ta seadme ühendada vajutades menüü nupule ning avanenud menüüs vajutada „Ühendused“, mille peale avaneb saadavalolevate seadmete nimekiri.

Kasutusjuht 2: Lülita tablo sisse või välja

Kasutaja saab mobiilirakenduses LED-tablood sisse või välja lülitada. Eelduseks on see, et nutiseade on juba LED-tablooga ühendatud. Tablo sisse- või väljalülitamiseks peab kasutaja ekraani paremal üleval servas vajutama menüü nupule. Menüü avanedes on kõige ülemises reas lüliti nupp, mida vajutades lülitub LED-tablo kas sisse või välja. Kui tablo ei ole parajasti ühendatud, on lüliti nupp ebaaktiivne.

Kasutusjuht 3: Kuva tablool sisu

Kasutaja saab LED-tablool kuvada tekste, visuaale või animatsioone. Iga sisu vormi jaoks on mobiilirakenduses eraldi lehekülg. Kui kasutaja on nutiseadme LED-tablooga ühendanud, avaneb talle vaade, kus on võimalik valida kolme lehekülje vahel. Visuaali kuvamiseks peab kasutaja ekraani all servas vajutama vasakpoolsele või parempoolsele nupule. Seejärel avaneb talle vastavalt kas eelloodud visuaalide vaade või kasutaja poolt loodud visuaalide vaade. Mõlemas vaates on visuaalid kuvatud kahe reana. Kui kasutaja on sobiva visuaali välja valinud, peab ta sellele peale vajutama ning see kuvatakse LED-tablool. Samuti ilmub mobiilirakenduse ülemisse äärde sama visuaal, mille kasutaja valis, et näidata viimati tabloole saadetud visuaali, juhul kui kasutaja ise LED-tablood parajasti ei näe.

Teksti kuvamiseks tablool peab kasutaja vajutama ekraani all servas asuvale keskmisele nupule, mille järel avaneb talle teksti vaade. Seal on teksti kuvamiseks kaks valikut. Esimene valik on vajutada mõne olemasoleva teksti peale, mille järel saadetakse see tabloole. Teine valik on kirjutada vaate all servas olevasse lahtrisse sobiv sõnum, valida värv, ning seejärel vajutada nuppu „Saada“, mille peale kirjutatud sõnum saadetakse tabloole. Ka saadetud tekst kuvatakse mobiilirakenduse üleval servas asuvale virtuaalsele tabloole.

Animatsioonide kuvamiseks LED-tablool peab kasutaja navigeerima eelloodud või kasutaja poolt loodud visuaalide lehele ning seejärel lehe üleval servas vajutama nupule

„Animatsioonid“. Kui kasutaja on avanenud loendis animatsiooni välja valinud, siis sellele vajutades saadetakse see LED-tabloole ning see kuvatakse ka rakenduses virtuaalsel tablool.

Kasutusjuht 4: Muuda tabloo heledust

Kasutaja saab mobiilirakenduses muuta LED-tablo heledust. Eelduseks on ühendus nutiseadme ning tablo vahel. Heleduse muutmiseks peab kasutaja vajutama rakenduse paremal üleval servas asuvale menüü nupule. Avanenud menüüs on teisel real liugur, millel sõrmega libistades muudab tablo heledust. Kui LED-tablo ei ole ühendatud, on liugur ebaaktiivne.

Kasutusjuht 5: Muuda tablo orientatsiooni

Kasutaja saab mobiilirakenduses muuta LED-tablo orientatsiooni. See võib olla vajalik, kui LED-tablo on kuskile üles pandud, aga tablo kuvab sisu valepidi. Orientatsiooni muutmise eelduseks on LED-tablo ühendus nutiseadmega. Orientatsiooni muutmiseks peab kasutaja vajutama rakenduse paremal üleval servas asuvale menüü nupule ning menüü avanedes vajutama ringja noolena kujutatud orientatsiooni muutmise nupule. Nupule vajutades LED-tablo orientatsioon vahetub.

Kasutusjuht 6: Loo sisu

Kasutaja saab mobiilirakenduses luua isiklikke visuaale, sõnumeid või animatsioone, mis salvestatakse nutiseadmesse. Loodud sisu on võimalik LED-tablool kuvada. Sisu loomiseks ei pea LED-tablo olema nutiseadmega ühendatud. Et sisu luua, peab kasutaja vajutama rakenduse all servas keskmisele teksti nupule või parempoolsele kasutaja poolt loodud visuaalide nupule.

Teksti lehel peab kasutaja vajutama lehe alumises paremas servas asuvale pluss-sümboliga nupule. Seejärel avaneb uute sõnumite loomise lehekülg, kus on erinevaid valikuid teksti loomiseks. Kasutaja saab lahtrisse kirjutada teksti, valida selle värvi ning valida, kas tekst kuvatakse staatiliselt või mööda tablood kerides. Kui loodud sõnum on valmis, peab kasutaja vajutama nupule „Salvesta“.

Visuaalide lehel peab kasutaja samuti lehe alumises servas vajutama pluss-sümboliga nupule. Sõltuvalt sellest, kas kasutaja on parasjagu visuaalide või animatsioonide alamlehel, avaneb talle visuaali või animatsiooni loomise aken. Visuaali loomise aknas on LED-tablood kujutav lõuend, millele kasutaja saab sõrmega joonistada endale sobiva

visuaali. Kasutaja saab valida lehel joonistamiseks sobivaid värve. Visuaali valmimisel peab kasutaja vajutama „Salvesta“, et loodud visuaal nutiseadmesse talletuks.

Animatsioonide lehel on sarnane LED-tablood kujutav lõuend, nagu visuaalide lehel. Erinevus on see, et lehe üleval servas on kuvatud joonistatavad visuaalid kaadritena. Animatsiooni loomiseks joonistab kasutaja sõrmega valmis ühe kaadri ning valib seejärel järgneva kaadri, mille peale avaneb talle uuesti tühi lõuend. Kui kaadreid on joonistatud piisavalt, peab vajutama nupule „Salvesta“, et animatsioon nutiseadmesse talletada.

Kasutusjuht 7: Muuda loodud sisu

Kasutajal on võimalik mobiilirakenduses enda poolt loodud sisu ehk sõnumeid, visuaale ja animatsioone muuta. Selleks peab ta valima ta teksti või visuaalide lehel valima mõne teksti või visuaali, mida ta sooviks muuta. Pärast muudetava sisu valimist peab kasutaja sellele peale vajutama ning talle avaneb valik sisu muutmiseks. Seejärel avaneb talle sama aken, mis sõnumi või visuaali loomisel, kuid see on juba täidetud sisuga, mida sooviti muuta. Kui muudatused on tehtud, peab kasutaja vajutama nuppu „Salvesta“, et need talletada.

Kasutusjuht 8: Vaata ja kasuta eelloodud sisu

Kasutajal on mobiilirakenduses võimalik sirvida eelloodud visuaale ja animatsioone, mis ei ole tema enda poolt loodud. Selleks peab kasutaja ekraani all servas vajutama vasakpoolsele nupule, mis avab eelloodud visuaalide vaate. Seal vaates on kahes reas saadaval erinevad eelloodud visuaalid või animatsioonid.

Kasutusjuht 9: Muuda rakenduse keelt

Kasutaja saab muuta mobiilirakenduse keelt, mille tagajärjel muutuvad ka kõik rakenduses nähtavad tekstid. Keele muutmiseks on kasutajal kaks võimalust. Esimeseks võimaluseks on avalehel olles vajutada ekraani ülemises paremas nurgas asuvat nuppu, mis avab modaalakna koos loeteluga keeltest. Iga valitava keele nimi on toodud vastavas keeles. Keele valimiseks vajutab kasutaja sobivale keelele ning rakendus muutub vastavalt. Teine võimalus on teistel lehtedel olles vajutada menüü nupule, mis asub samuti ekraani paremal üleval servas. Avanenud külgmenüüst tuleb vajutada nupule keel, mille järel avaneb keeleveliku modaalaken. Valinud sealt sobiva keele, muutub rakendus vastavalt.

8.2 Arhitektuur

Loodud mobiilirakendus koosneb erinevatest komponentidest, kuvadest, laost ning teenustest, mis asuvad rakenduse juurkaustas nimetustele vastavates kaustades „*components*“, „*app*“, „*store*“ ning „*services*“. Lisaks on juurkaustas kaust „*assets*“, mis sisaldab rakenduse ressursse ehk kasutatavaid pilte, kirjagarnituure, ikoone ja kaust „*locales*“, mis sisaldab rakenduse keelefaile JSON (*JavaScript Object Notation*) andmevormingus. Kasutaja näeb rakenduse kuvasid, mis sisaldavad endas väiksemaid komponente ning kasutavad teenuseid. Kasutatavad teenused on keeleteenus ning teavitusteenus. Keeleteenus tuleb teegist „*i18next*“ ning võimaldab pakkuda mitmekeelsust terves rakenduses. Teavitusteenus on loodud, et kuvada sarnaseid teavitusi erinevates rakenduse komponentides. Rakenduses kasutakse Redux teegi poolt pakutavat ladu, et hoiustada globaalseid muutujaid, mis oleks kõigis komponentides kättesaadavad ja muudetavad. Komponentides kontrollitakse, kas laos olevate muutujate väärtused on muutunud ning selle põhjal valitakse, kas värskendada komponendi välimust või mitte. Ladu kasutab omakorda teeki „*Redux Persist*“, et hoiustada neid globaalseid muutujaid võti-väärtus paaridena nutiseadme püsimälus.

Kõigist kuvade-, teenuste- ning komponentidevahelistest seostest parema ülevaate saamiseks loodi mobiilirakenduse komponentdiagramm, mis on saadaval Lisas 3.

Mobiilirakendus kasutab kuvade vahel navigeerimiseks Expo ruuterit, mis põhineb failipõhisel marsruutimisel. Selleks on rakenduse juurkaustas kaust nimega „*app*“, kuhu mingi komponendi (kuva) lisamisel muudetakse see automaatselt üheks marsruudiks. See tähendab seda, et soovitud kuvale pääsemiseks peab ruuterile edastama vaid faili asukoha, pole vahet, kui sügaval rakenduses viibida [50]. Antud süsteem on mugav, sest tänu sellele saab hea ülevaate kogu rakenduse struktuurist ka ainult kaustasid vaadates.

8.3 Ladu

Lao eesmärk on React Native mobiilirakenduses hoiustada globaalseid väärtuseid, millele oleks ligipääs kõigist rakenduse komponentidest. Lisaks väärtustele sisaldab ladu ka ligipääsetavaid funktsioone (tegevusi) nende väärtuste muutmiseks. Ladu pakub mobiilirakenduses teeki nimega „*Redux*“ ning laole vastab koodis fail „*store.js*“. Antud rakenduse puhul on ladu jagatud kolmeks osaks (viiluks), milles hoiustatakse erinevate

rakenduse osade muutujaid. Nendeks osadeks on „matrix“, „ble“ ja „device“, mis hoiustavad vastavalt nimedele LED-tablooga (mikrokontroller), Bluetooth LE ühendusega ning seadmega seotud väärtusi. Kõigist laos olevatest muutujatest ülevaate andmiseks loodi Joonis 9.

BluetoothState	MatrixState	DeviceState
<pre>- allDevices: DeviceReference[] - currentData: string[] - connectedDevice: DeviceReference null - retrievedData: string null undefined - bluetoothEnabled: boolean null + setDevice(device: DeviceReference): void + setConnectedDevice(device: DeviceReference): void + setRetrievedData(data: string null undefined): void + setBluetoothState(state: boolean null): void</pre>	<pre>- isOn: boolean - myDesigns: string[][] - myAnimations: string[][][] - presets: string[][] - currentMatrix: string[] - noOfFrames: number - color: string - brightness: number + addToMyDesigns(design: string[]): void + addToPresets(preset: string[]): void + addToMyAnimations(animation: string[][]): void + removeFromMyDesigns(design: string[]): void + setCurrentDesign(design: string[]): void + setCurrentFrames(frames: number): void + setCurrentColor(color: string): void + setCurrentState(state: boolean): void + setCurrentBrightness(brightness: number): void</pre>	<pre>- orientation: string - screenWidth: number - screenHeight: number</pre>

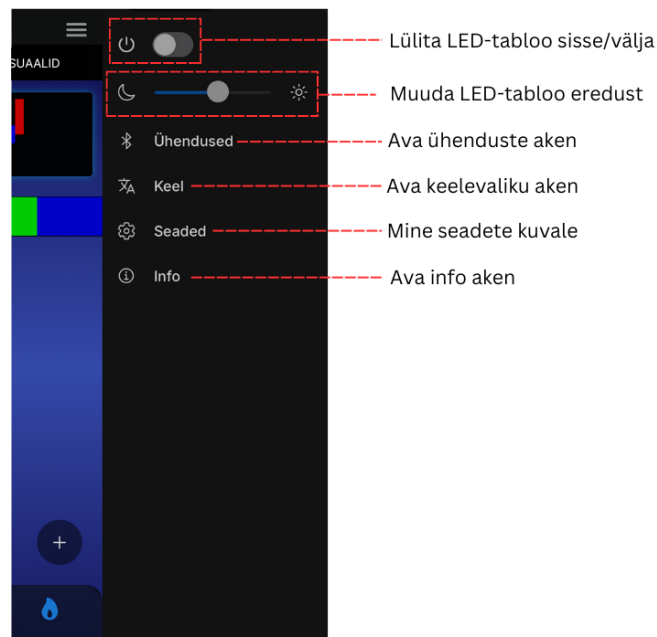
Joonis 9. Mobiilirakenduse lao osad ning osades paiknevad väärtused ja funktsioonid.

8.4 Komponentid

Järgnevalt tuuakse välja mõned mõned olulisemad loodud mobiilirakenduse komponendid ning kirjeldatakse nende tööpõhimõtet.

8.4.1 Külgmenüü

Külgmenüü komponent on mobiilirakenduses kasutusel, et oleks lihtne ligipääs LED-tablooo ereduse ning oleku muutmiseks, ühenduse haldamiseks, keele muutmiseks, rakenduse teabe vaatamiseks ning rakenduse seadete avamiseks (vt Joonis 10). Külgmenüü on realiseeritud kasutades Expo poolt pakutavat „Drawer“ komponenti [51].



Joonis 10. Mobiilirakenduse külgmenüü komponent.

Komponenti saab avada kuvades „Minu kujundused“, „Tekst“ ning „Eelloodud kujundused“, vajutades ekraani üleval paremal nurgas asuvat kolme paralleelset joont kujutava sümboliga menüünuppu. Nimetus „külgmenüü“ tuleneb sellest, et menüü avaneb paremalt küljelt ning katab ära vaid ühe külje ekraanist.

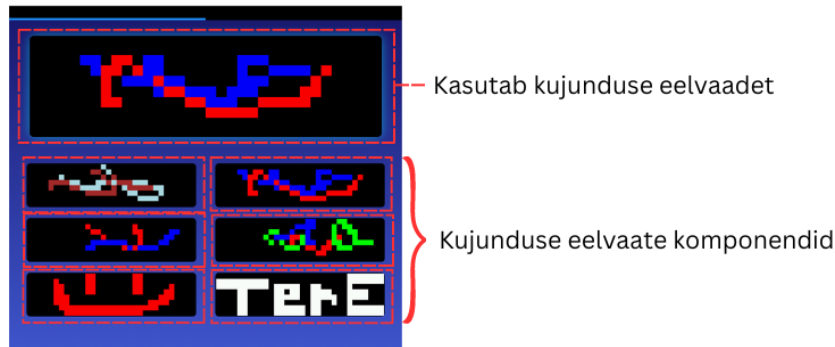
Külgmenüü komponendis kutsutakse LED-tablo lüliti vajutamisel välja funktsioon, mis muudab laos LED-tablo oleku väärtust ning saadab LED-tabloole käsu muuta kõigi pikslite värv mustaks ehk lülitada tablo välja. Sarnaselt toimib ka ereduse muutmise nihuti, millel nihutades muudetakse väärtust laos ning saadetakse LED-tabloole käsk ereduse muutmiseks.

Menüü on list viie erineva üksteise alla paigutatud funktsionaalsusega. Nendeks on, seadme info, keele muutmise, seadme lahti kinni ühendamine, heleduse muutmise ja miskit veel. Menüü avaneb paremalt ning on musta taustaga, valgete tekstidega ja vastavalt sellele, mis leheküljel seda komponenti avati, vastavalt sellele jääb eelnevalt avatud olnud lehekülge ka osaliselt vasakust servast näha.

8.4.2 Kujunduse eelvaade

Antud komponent võimaldab eelvaadet mobiilirakenduses loodud kujundusele, mille saab LED-tabloole saata. Kujunduse eelvaate komponendist on kaks variatsiooni: visuaali eelvaade ehk staatiline pilt, millele vastab koodis „DesignPreview.jsx“ ning animatsiooni

eelvaade ehk liikuv pilt, millele vastab „AnimationPreview.jsx“. Komponenti kasutatakse omakorda teistes komponentides (vt Joonis 11), nagu visuaalide ja animatsioonide loetelud („DesignList.jsx“, „AnimationList.jsx“), animatsiooni kaadrid („AnimationFrames.jsx“) ning hetkekujunduse komponent („CurrentMatrix.jsx“).



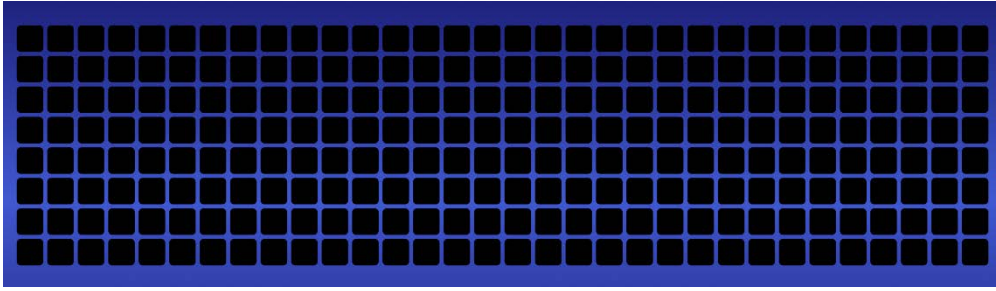
Joonis 11. Mobiilirakenduse kujunduse eelvaate komponendi erinevad kasutused.

Loeteludes kasutatakse kujunduse eelvaate komponenti, et kuvada kõigi parameetrikus antud massiivis leiduvate kujunduste eelvaated. Animatsiooni kaadrite komponendis kasutatakse kujunduse eelvaadet, et animatsiooni loomisel näidata juba loodud kaadreid. Hetkekujunduse komponendis näidatakse kujunduse eelvaadet sellest kujundusest, mis viimati edukalt LED-tabloole saadeti.

Visuaali eelvaate komponent nõuab parameetriteks massiivi, kus iga väärtus on loodud kujunduse ühe piksli värv, ja laiust, mis määrab ära eelvaate laiuse. Eelvaate loomiseks läbitakse värvide massiiv ning igast värvist luuakse ruuduke, mis sümboliseerib ühte pikslit, millest kokku moodustub saadud laiuse alusel SVG (*Scalable Vector Graphics*) vormingus pilt. Laiuse valiku vajadus on tingitud komponendi kasutamisest teistes eri suurustega komponentides.

8.4.3 Lõuend

Lõuendi komponendi eesmärk on lubada kasutajal joonistada kujundus, mida ta sooviks näha LED-tablool. Komponentile vastab koodis failinimi „DrawingPadGrid.jsx“. Antud komponenti kujutatakse mustadest pikslitest koosneva ruudustikuna (vt Joonis 12).



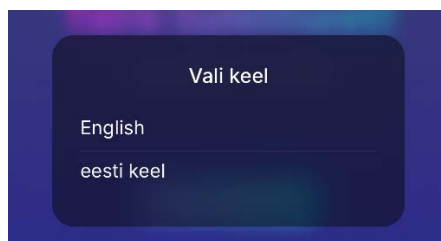
Joonis 12. Mobiilirakenduse lõuendi komponent LED-tabloole vastava kuvasuhte ning pikslite arvuga.

Lõuendi kujutusviisiks valiti ruudustik, mitte tühi leht, kuna ühest ruudust väiksemat ühikut ei ole võimalik tablool kuvada ning lisaks annab ruudustik autori arvates kõige parema ettekujutuse, milline loodud visuaal LED-tablool lõpuks välja võib näha. Ruudustiku pikslite arv, kõrgus ja laius sõltub rakendusega kasutatava LED-tablooo mõõtmetest. Kuna arenduse käigus kasutati 8x32 resolutsiooniga tablood, siis on sama resolutsioon vaikumisi mobiilirakendusse sissekirjutatud ka lõuendi suuruseks. Dünaamiline lõuendi suuruse muutumine sõltuvalt parajasti ühendatud LED-tabloost on lõputöö võimalik edasiarendus.

Komponent kasutab toimimiseks hetkel valitud värvi, mis saadakse laost, mis omakorda asetatakse lattu kas värvivaliku modaalakna komponendist või hiljutiste värvide komponendist. Iga ruudu värv on hoiustatud massiivis, kus iga element vastab ühele ruudule. Iga ruut on ka eraldi komponent, millele vastab mingi kindel koordinaatide vahemik ekraanil. Kui kasutaja nutiseadme ekraani puudutab, leitakse, kas puutepunkti koordinaadid jäid mõne ruudu koordinaatide vahemikku, ning kui see tingimus oli täidetud, siis värvitakse ruut valitud värviga. Vastasel juhul puudutust ignoreeritakse.

8.4.4 Keelevaliku modaalaken

Antud komponendi eesmärk on pakkuda kasutajale võimalust rakenduse keele valikuks. Komponendile vastab koodis failinimi „LanguageModal.jsx“. Komponent kujutab endast ekraani keskel avanevat akent, kus on loetelu erinevatest rakenduses saadaval olevatest keeltest (vt Joonis 13).



Joonis 13. Mobiilirakenduse keeleveliku modaalaken.

Hetkel on valikus kaks keelt: eesti keel ja inglise keel. Iga keel on välja toodud vastava riigi emakeeles, et kasutajal oleks lihtne õiget keelt leida. Keele valimiseks tuleb vajutada nimekirjas sobiva keele peale, mille järel keeleteenus muudab rakenduse keele.

8.4.5 Ühenduste modaalaken

Antud komponendi eesmärk on kuvada kasutajale saadaolevaid seadmeid, võimaldada tal nende seadmetega ühendada, lahti ühendada ning näha ühenduse staatust. Komponentile vastab koodis failinimi „ConnectionsModal.jsx“. Komponent kujutab endast akent, mis avaneb ekraani keskel ning sisaldab endas loetelu Bluetooth'i abil leitud seadmetest, nende staatusest ning seadmete otsimise staatusest (vt Joonis 14). Iga leitud seadme kõrval on nupp, mis võimaldab nutiseadet sellega ühendada või sellest lahti ühendada. Korraga võimaldab mobiilirakendus hetkel ühendada ühe LED-tabloo, seega kui ühe tablooga on ühendus tehtud, siis uute seadmete otsimine lõpetatakse.



Joonis 14. Mobiilirakenduse ühenduste modaalaken.

Saadaolevad seadmed ning ühenduste staatused võetakse mobiilirakenduse laost ning lattu asetatakse need teegi „react-native-ble-plx“ abil loodud Bluetooth-ühenduse funktsioonide tulemuste alusel [41]. Seadme kõrval nupule „Ühenda“ või „Ühenda lahti“ vajutamisel kutsutakse välja samuti üks eelnevalt nimetatud teegi abil loodud funktsioon, mis ühendab, või ühendab lahti nutiseadme valitud seadmega nimekirjast.

8.4.6 Värvivaliku modaalaken

Antud komponent võimaldab kasutajal valida värvi, mida kasutada kas värvilise teksti loomiseks või kujunduste joonistamiseks. Komponentile vastab koodis failinimi „ColorPickerModal.jsx“. Komponent kujutab endas akent, mis avaneb ekraani keskele ning sisaldab endas värvide eelvaadet, ringi, millel on kujutatud värvispekter, heleduse nihuti ning nuppu „OK“. Värvispektri ring on puutetundlik, seega saab kasutaja värvispektril meeldivale värvile vajutades värvi valida. Mobiilirakenduses hetkel kasutusel olev värv ja uus värv on kujutatud värvide eelvaates modaalakna ülemises servas, kus on välja toodud ka värvide HEX-kood. Heleduse nihuti paikneb ringikujulise värvispektri all ning võimaldab valida valitud värvi heledust. Värvivaliku kinnitamiseks peab vajutama nupule „OK“, vastasel juhul värvi ei muudeta. Värvispekter, värvide eelvaade ning heleduse nihuti pärinevad teegist „reanimated-color-picker“ [52].

8.5 Teenused

Teenused on taaskasutatavad rakenduse osad, mis sisaldavad endas ärioloogikat või andmeid, mida on võimalik kasutada mitmes erinevas komponendis. Iga teenus sisaldab endas tavaliselt mingi kindla valdkonnaga seotud loogikat. Loodud mobiilirakenduses on kokku kolm teenust: keeleteenus, teavitusteenus ning teenus „hexToBitmap“, mis sisaldab funktsioone, mida kasutatakse lõuendi komponendiga loodud kujunduste värvimassiivide töötlemiseks.

8.5.1 Keeleteenus

Keeleteenust kasutatakse rakenduse mitmekeelsuse saavutamiseks. Teenuse failinimi on rakenduse koodis „i18next.js“. Antud teenust kasutatakse, et „i18next“ teegi poolt pakutud lokaliseerimise teenus vastavalt keelefailides saadaolevatele keeltele seadistada ning valitud seadistusega kogu rakenduses saadavaks teha. Algseadistamisel määratakse vaikimisi keel, milleks on eesti keel ning varukeel, milleks on inglise keel. Varukeelt kasutatakse siis, kui mõni sõne on jäänud valitud keelde tõlkimata, kuid seda on vaja siiski kuvada.

8.5.2 Teavitusteenus

Teavitusteenuse eesmärk on võimaldada valitud eelloodud teavitusi erinevates mobiilirakenduse komponentides taaskasutada. Teenusele vastab koodis failinimi

„AlertService.js“. Teavituse all on silmas peetud modaalakent, mis avaneb mingi sündmuse korral ekraani keskel muu sisu kohale ning annab kasutajale infot, või sunnib kasutajat midagi tegema.

Antud teenus sisaldab hetkel kahte funktsiooni: „showBluetoothAlert“ ja „showDisconnectAlert“. Esimene funktsioon kuvab kasutajale teate, et Bluetooth ei ole seadmes aktiveeritud, või Bluetooth'i seadetes ei ole lubatud uued Bluetooth-ühendused. Teate modaalaknal on kaks nuppu: „Seaded“ ja „Tühista“. Vajutades nupule „Seaded“ suunatakse kasutaja nutiseadme seadetesse, et ta saaks Bluetooth'i sisse lülitada või uued ühendused lubada. Vajutades „Tühista“ antud teade suletakse.

Teine funktsioon „showDisconnectAlert“ kuvab ekraanil modaalakna, mis teavitab kasutajat seadme (LED-tabloo) lahti ühendamise korral. Antud funktsioon on vajalik, et kasutaja saaks kohe teada, kui ühendus katkeb, mitte ei peaks seda hiljem rakenduse kasutamise käigus avastama.

8.5.3 hexToBitmap

Antud teenuse eesmärk on pakkuda rakendusele funktsioone, mis võimaldavad teisendada HEX-koodidest koosnevat massiivi, mis tekib lõuendi komponendis värvide kujunduse salvestamisel. Teenusele vastab koodis failinimi „hexToBitmap.js“. Teenust on vaja, sest loodud värvikoodide massiiv ei ole sobivas vormingus, et seda saaks kuvada kujunduse eelvaate komponendis, või edastada andmetena Bluetooth'i abil LED-tabloole.

Teenuses on kaks funktsiooni: „hexArrayToBitmap“ ja „hexArrayToString“. Esimene funktsioon muudab HEX-koodide massiivi sobivaks massiiviks, mida saab kasutada kujunduse eelvaate loomisel. Kui esialgu on massiivis väärtused sõnedena kujul „#FF0000, #00FF00“, siis pärast teisendust saadakse kuueteistkümnendikarvuline kuju „0xF800, 0x07E0“, mis on sobilik kujunduse eelvaate loomiseks.

Teine funktsioon muudab HEX-koodide massiivi üheks pikaks sõneks, käies terve massiivi läbi ning lisades iga järgmise väärtuse sõne lõppu. Esialgne formaat „#FF0000, #00FF00“ teisendatakse seega formaadiks „FF000000FF00“. Antud teisendus on vajalik, sest hetkel on nii mobiilirakenduses andmete saatmiseks kui ka sardtarkvaras andmete vastuvõtmiseks lihtsuse huvides kasutusel sõne andmetüüp, seega ei ole massiiv sobilik.

8.6 Kuvad

Kuvadeks kutsutakse React Native'i rakenduses erinevaid lehekülgi või vaateid. Need omavad faililaiendit `-.jsx` või `-.tsx` ning on peamiselt graafilised komponendid, mis sisaldavad endas väiksemaid sõltuvaid või sõltumatuid komponente ning kirjeldavad HTML'i ja CSS'iga lahti, kuidas leheküljel erinevad osad paiknema peavad ning millised välja nägema [53]. Et need erinevad osad reageeriks ja muutuks vastavalt kasutaja tegevusele, on kuvades kasutusel ka JavaScript.

Kuvade loomisel alustati makettide koostamisest Canva's. Canva on tasuta kasutatav graafilise disaini platvorm, mis pakub mugavat mallipõhist kujunduste loomise võimalust [54]. Selle keskkonnaga oli autor ka varasemalt kokku puutunud erinevate plakatite kujundamisel, seega ei esinenud makettide loomisel raskusi (vt Joonis 15).



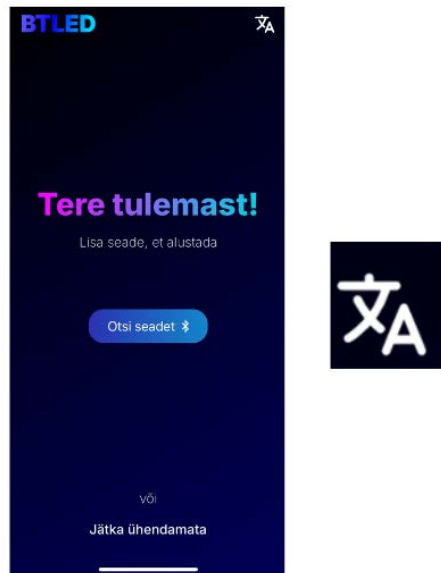
Joonis 15. Canva keskkonnas loodud maketid mobiilirakenduse kuvadele.

Kokku loodi arenduse käigus seitse erinevat kuva. Järgnevad alapeatükid kirjeldavad kõiki loodud mobiilirakenduse kuvasid lähemalt, tuues välja nende eesmärgi, kasutuse ning eripärad.

8.6.1 Avaleht

Avaleht on kuva, mis avaneb rakenduse esmakäivitamisel. See on vähese informatsiooniga ja kompaktne, et kasutajal oleks lihtne rakenduse kasutamist alustada. Koodis vastab sellele kuvale fail `„index.jsx“`. Kuva eesmärk on kasutajat tervitada ning suunata teda LED-tablooga ühendama või rakendust kasutama. Avalehel on rakenduse

nimi, tervitav sõnum ning kolm erinevat nuppu, millele kasutaja saab vajutada. Lehe ülemises nurgas on nupp, millele vajutades avaneb modaalaken, kus on võimalik rakenduse keelt valida. Hetkel saab valida kahe keele vahel – eesti või inglise. Nuppu kujutab üheselt mõistetav sümbol avalehel, et esmakasutajatel oleks seda lihtne leida ja ei peaks hakkama hiljem juba rakendust kasutades keelevelikut otsima (vt Joonis 16).

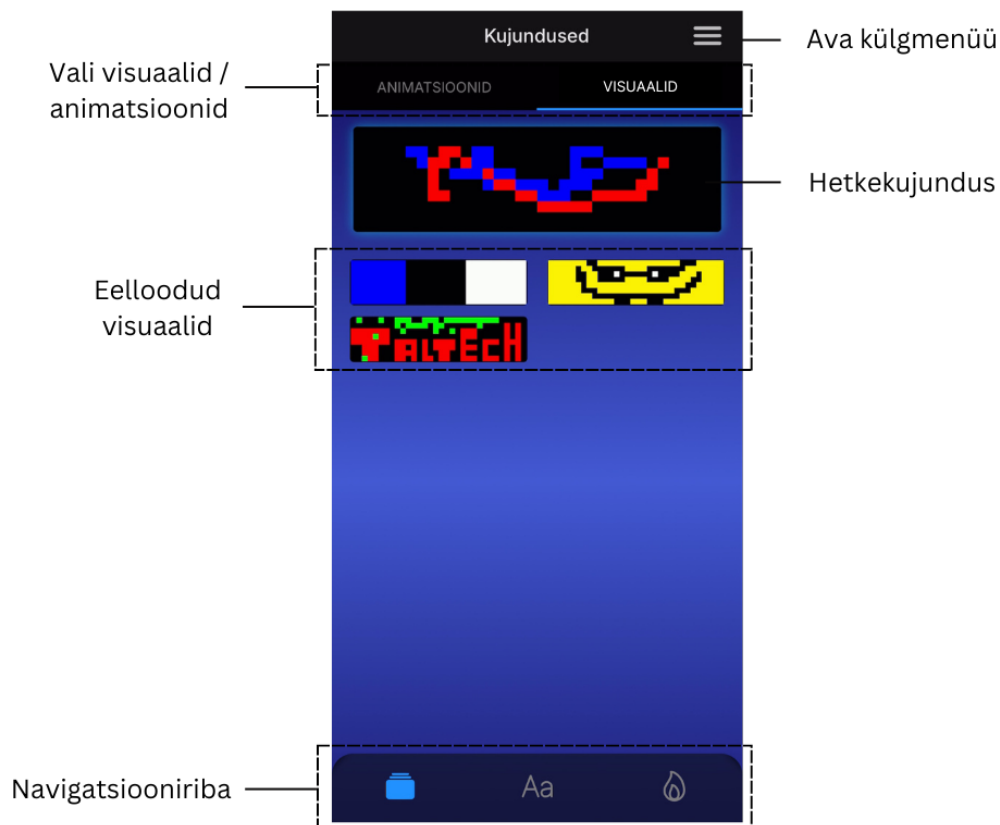


Joonis 16. Mobiilirakenduse avaleht ning lähivaade keeleveliku nupule.

Teiseks nupuks avalehel on “Otsi seadet” koos Bluetooth’i sümboliga. Nupp on paigutatud suurelt ekraani keskele, et see kohe kasutajale silma hakkaks. Antud nupule vajutades avaneb modaalaken saadaolevate Bluetooth-seadmetega, mida mobiilirakendus toetab. Hetkel on seadmed filtreeritud nime alusel ehk rakendus otsustab ennast reklaamiva LED-tablo nime põhjal, kas see sobib ja saab seda kasutajale näidata või mitte. Kui sobiv seade on leitud ja sellega ühendatud, suunatakse kasutaja modaalakna sulgemisel rakenduse peakuvale, kus saab näha eelloodud visuaale. Kui kasutaja ei soovi, või tal pole võimalik siiki nutiseadet kohe LED-tablooga ühendada, on avalehe alumises servas alternatiiviks ka peenikeses kirjas nupp “Jätka ühendamata”. Sellele nupule vajutades suunatakse kasutaja kohe eelloodud kujunduste lehele. Ühendamata olekus saab kasutaja rakenduses erinevaid visuaale luua, muuta, vaadata või kustutada, kuid LED-tabloole saatmine pole võimalik. Kui kasutaja soovib siis seadmega ühendada, on tal küljemenüü avamisel seda võimalik teha.

8.6.2 Eelloodud kujundused

Eelloodud kujunduste kuva eesmärk on näidata kasutajale erinevaid visuaale ning animatsioone, mida selle rakendusega võimalik luua on, ning lubada kasutajal neid kujundusi ka LED-tabloole saata. Antud kuva koosneb kahest sektsioonist – animatsioonid ja visuaalid, mille vahel on võimalik valida sõrmega paremale või vasakule libistades (vt Joonis 17). Sektsioonidele vastavad koodis failinimed „presets.jsx“ ning „animations.jsx“.

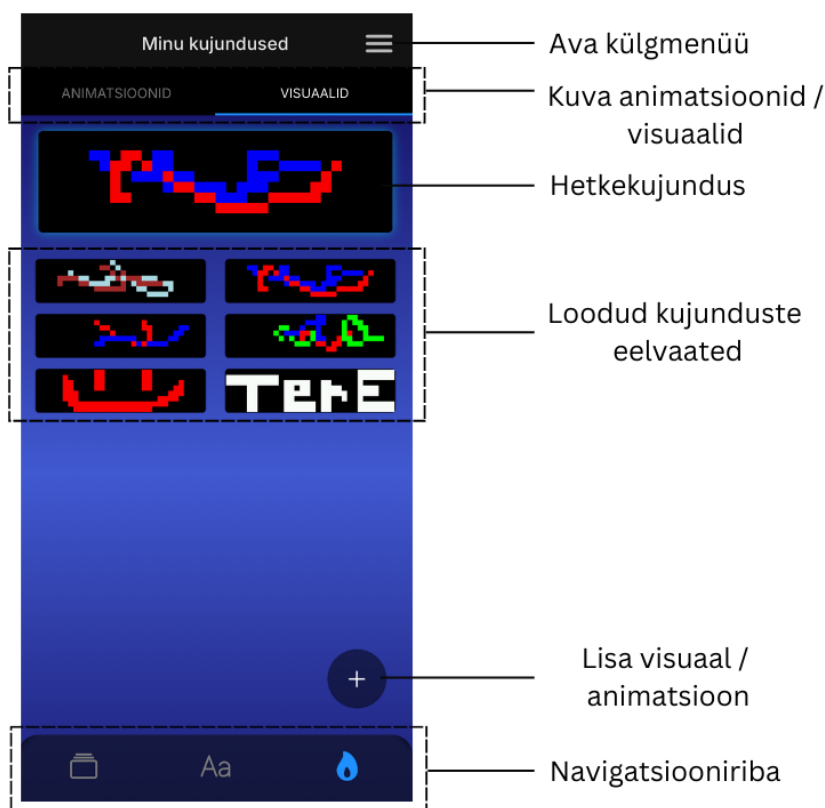


Joonis 17. Mobiilirakenduse kuva „Eelloodud kujundused“ visuaalide sektsioon.

Eelloodud kujunduste lehe mõlemad sektsioonid sisaldavad endas järgnevaid komponente – alumine navigatsiooniriba, mis võimaldab kasutajal liikuda erinevate kuvade vahel; menüü nupp – mis võimaldab avada külgmenüü; hetkekujunduse komponent, mis näitab kasutajale, millist kujundust hetkel või viimati LED-tablool kuvati; eelvaadete loetelu eelloodud kujundustest, mis võimaldab kasutajal valida mingi kujunduse, mida LED-tabloole saata. Eelloodud kujunduste loetelust visuaali või animatsiooni LED-tabloole saatmiseks peab kasutaja lihtsalt sobivale kujundusele vajutama ning valida valikutest saatmise sümboli.

8.6.3 Minu kujundused

Antud kuva eesmärk on näidata kasutajale kujundusi, mis on tema enda poolt loodud, ning võimaldada neid LED-tabloole saata. Lisaks võimaldada kasutajal luua uusi kujundusi või muuta ja kustutada olemasolevaid. Minu kujunduste kuva koosneb sarnaselt eelloodud kujunduste kuvale kahest sektsioonist: animatsioonid ning visuaalid, mille vahel on võimalik valida sõrmega libistades (vt Joonis 18). Animatsioonidele vastab koodis failinimi „myanimations.jsx“ ning visuaalidele „mydesigns.jsx“.



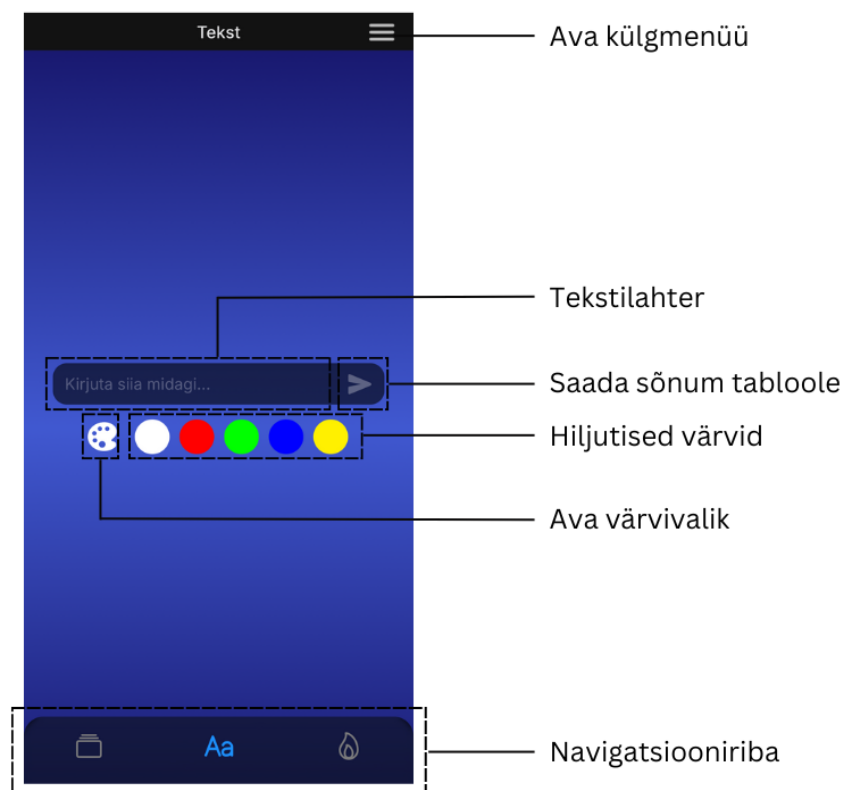
Joonis 18. Mobiilirakenduse kuva „Minu kujundused“ visuaalide sektsioon.

Leht on peaaegu identne eelloodud kujunduste lehele ehk kõige üleval on hetkekujunduse komponent ja menüü nupp, lehe keskel valik loodud kujunduste eelvaadetest ning all servas navigatsiooniriba. Loodud kujundustule vajutades on võimalik neid muuta, kustutada, või kuvada LED-tablool. Erinevaks teeb kuva see, et lehe alumises paremas nurgas on pluss-sümboliga nupp, mida tuntakse kui hõljuv tegevuse nupp (inglise keeles *floating action button*), mille nimi tuleneb sellest, et see hõljub tavaliselt ülejäänud sisu kohal ning võimaldab vajutamisel mingit tegevust [55]. Nupule vajutades suunatakse kasutaja kas uue animatsiooni või uue visuaali loomise lehele, vastavalt sellele, millises sektsioonis

kasutaja hetkel viibib. Navigatsiooniribal on antud kuva tähistatud tuleleegi sümboliga, näitamaks, et sellel kuval midagi küpseb.

8.6.4 Tekst

Antud kuva pakub kasutajale võimalust kirjutada sõnum ning saata see LED-tabloole kuvamiseks. Lisaks on võimalik valida ka mõne eelloodud sõnumi vahel (vt Joonis 19).

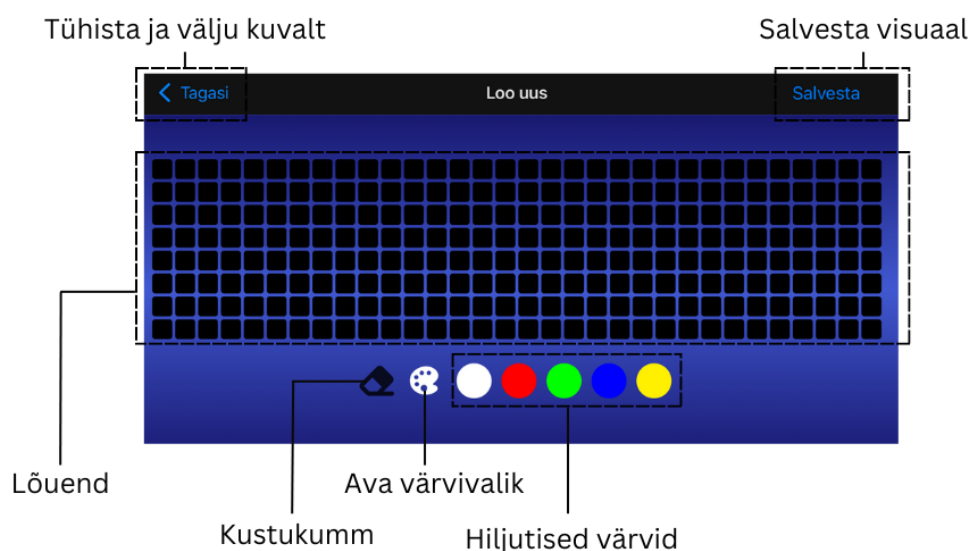


Joonis 19. Mobiilirakenduse kuva „Tekst“.

Programmikoodis vastab kuvale failinimi „text.jsx“. Lehe ülemises servas on menüü ning keskel tekstilahter, kuhu kasutaja saab oma sõnumi kirjutada. Tekstilahtri kõrval on noole sümboliga nupp sõnumi LED-tabloole saatmiseks. Tekstilahtri all on ka värvivaliku ikoon, mis avab sellele vajutades modaalakna, kus saab valida meelepärase värvi, millega sõnum ekraanil kuvatakse. Ikooni kõrval on viis viimati kasutatud värvi kiireks ligipääsuks. Kuva alumises servas on navigatsiooniriba.

8.6.5 Loo visuaal

Antud kuva võimaldab kasutajal luua või muuta visuaali. Kuva eripäraks on see, et see avatakse horisontaalses asendis, seega peab kasutaja oma nutiseadet pöörama, et sellel kuval tegutseda. Horisontaalse asendi eesmärk on pakkuda visuaali loomiseks suuremat ekraanipinda, kuna eeldatud on, et LED-tablood kasutatakse horisontaalasendis (vt Joonis 20).



Joonis 20. Mobiilirakenduse kuva „Loo visuaal“.

Visuaali loomise kuva peamine komponent on lõuend, millele kasutaja saab sõrmega joonistada sobiva visuaali, mida ta sooviks LED-tablool näha. Lõuendi komponendi all paiknevad kustutamisenupp ning värvivaliku nupp. Kustutamisenupule vajutades aktiveerub kustukumm ning kui kasutaja pärast seda lõuendil mõnele ruudule vajutab, siis see tühjendatakse. Värvivaliku nupule vajutades avaneb värvivaliku modaalaken, kust kasutaja saab sobiva värvi valida, millega visuaali luua. Antud kuva ülemises ääres paiknevad kaks nuppu: vasakul pool „Tagasi“ ning paremal pool „Salvesta“. Kui kasutaja soovib visuaali loomisest loobuda, pääseb ta „Tagasi“ nupule vajutades „Minu kujundused“ lehele ning visuaali ei salvestata. Kui kasutaja soovib loodud visuaali LED-tablool näha, peab ta vajutama „Salvesta“, mille peale ta suunatakse „Minu kujundused“ lehele, kust tal on võimalus ühendatud LED-tablool korral loodud kujundust sellel näidata.

8.6.6 Loo animatsioon

Antud kuva on oma ülesehituselt sarnane visuaali loomise kuvale. Kuva avaneb samuti horisontaalses asendis, selle ülemises servas on nupud „Tagasi“ ja „Salvesta“, keskel on

lõuend, mida kujutab ruudustik ning selle all on kustutamise ja värvivaliku nupp (vt Joonis 21).



Joonis 21. Mobiilirakenduse kuva „Loo animatsioon“.

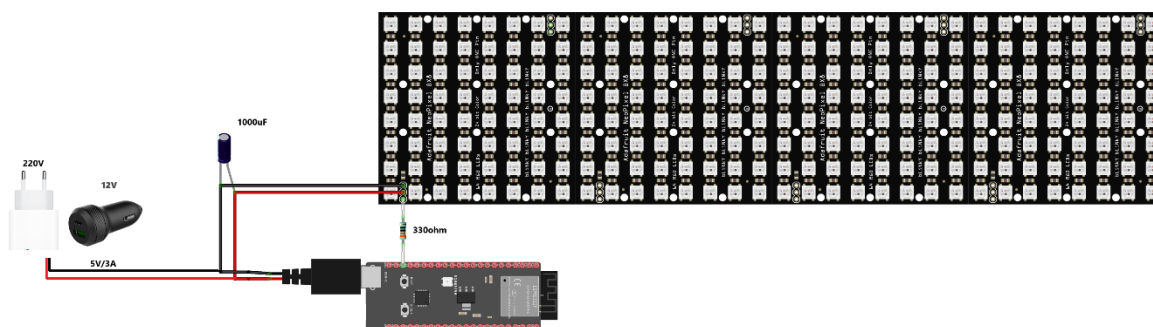
Erinevuseks on see, et kuna animatsioon koosneb üksikutest kaadritest, siis asub lõuendi ning lehe ülemise serva vahel kaadrite komponent. Kaadrite komponendil kuvatakse, mitmest kaadrist loodav animatsioon koosneb ning millist kaadrit parasjagu muudetakse. Komponendis saab uusi kaadreid lisada ja vanu eemaldada. Kui kaadreid on rohkem, saab ekraanist välja jäävaid kaadreid näha komponendil sõrmega paremale või vasakule viibates. Animatsiooni salvestamiseks peab kasutaja vajutama „Salvesta“ nupule, mille peale kujundus salvestatakse ning kasutaja suunatakse „Minu kujundused“ lehele.

8.6.7 Seaded

Seadete kuva on loodud selleks, et kasutajal oleks võimalus vajadusel muuta LED-tablo suurust ja muid omadusi, või seda, milline rakendus välja näeb. Käesoleva lõputöö arenduse käigus ei jõutud antud funktsionaalsusi veel luua, seega on antud kuva valik küll menüüs olemas, kuid visuaalset osa veel ei ole.

9 Lõpplahendus

Käesoleva lõputöö lahendusena valmis kaheosaline süsteem Bluetooth'i abil juhitud LED-tabloost ning React Native'i baasil iOS mobiilirakendusest. Süsteemi jaoks kirjutatud tarkvara on avatud lähtekoodiga, litsenseeritud GNU GPLv3 litsensi alusel ja kättesaadav GitHub hoidlas veebiaadressil <https://github.com/star2k1/ledmaster>. Valminud süsteem on loodud kasutamiseks sisetingimustes ning eeldusel, et hoones kasutamisel tuleb toide seinapistikust või sõidukis kasutamisel sigaretisüütajasse asetatavast adapterist. LED-tablo ja mikrokontroller kasutavad toitepingeks viite volti ning suurim kasutatav voolutarve on kolm amprit. Toite jaoks kasutatakse kaasaegset USB-C ühendust, mis tähendab, et vooluadapter või autos kasutatav pistik peab omama USB-C pesa ning suutma välja anda vähemalt kolm amprit, et süsteem ootuspäraselt toimiks (vt Joonis 22, Lisa 4).



Joonis 22. Lõpplahenduse üldine skeem.

Valminud mobiilirakendust saab kasutada iOS nutiseadmes, kuid hetkel ei ole see saadaval *App Store*'is, vaid tuleb lähtekoodist ise kompileerida. Rakenduse *App Store*'i paigutamine tähendaks rahalisi kulusi, seega otsustati antud lõputöö raames see vahele jätta. Mobiilirakendus võimaldab nutiseadme LED-tablooga ühendada ning seejärel saab nutiseadmest saata LED-tabloole kuvamiseks sõnumeid, visuaale ning animatsioone. Valminud mobiilirakenduses saab saatmiseks kasutada eelloodud kujundusi või luua ise midagi uut. Rakenduse kaudu saab LED-tablo eredust muuta ning seadet välja ja sisse lülitada.

10 Võimalikud edasiarendused

Antud lõputöö raames saadi peamised eesmärgid täidetud, kuid lisaeesmärgini, milleks oli loodud prototüübi kompaktseks tegemine, ei jõutud. Seega on üheks võimalikuks edasiarenduseks lõpplahenduse välimuse viimistlus, mis sisaldab endas arendusplaadile parema korpusse loomist, LED-tabloole ümbrise ning LED-idele valguse hajutamiseks katematerjali lisamist. Veelgi rohkem arendusega edasi minnes võiks ise disainida uue trükkplaadi, mis on mõeldud spetsiifiliselt taolist tüüpi juhtmevaba ühendusega LED-tabloodele. Lisaks võiks luua mingi viisi, et tablood oleks mugav erinevatele pindadele kinnitada.

Mobiilirakenduses arendati lõputöö raames välja vaid baasfunktsionaalsused, seega on võimalikke edasiarendusi palju. Esimene edasiarendus oleks toe lisamine ka mõnele teisele LED-tabloole, mida sardtarkvaras kasutatavates teekides toetatakse. See tähendab rakendusesisest seadistusvõimalust, kus kasutaja saab valida enda tabloo mõõtmed ning muud parameetrid. Lõputöö autor soetas juba ühe teise, HUB75-tüüpi tabloo, millega kavatsetakse järgnevalt arendama asuda.

Teine edasiarendus võiks olla mobiilirakendusse LED-tabloo jaoks suurema hulga funktsionaalsuste lisamine. Kui hetkel on võimalik küll erinevat tüüpi sisu kuvada (visuaalid, animatsioonid, tekst), siis ei paku rakendus väga valikut, kuidas seda sisu täpselt kuvatakse. Puuduvad eriefektid, kiiruse muutmine, tabloo pööramine, teksti individuaalsete tähtede värvimine ja nii edasi. Lisaks võiks kasutajal olla võimalus võtta pilt oma nutiseadme galeriist ning sisestada see rakendusse, mis muudab selle suurus vastavalt, et kuvada pilti LED-tablool. Suur osa nendest valikutest on LED-tabloole mõeldud teekides juba olemas, kuid tuleb panna toimima sardtarkvaras ning mobiilirakenduses.

Kuna rakendus on loodud avatud lähtekoodi põhimõttel, siis võiks kolmandaks edasiarenduseks olla kogukonna funktsionaalsuste loomine. See tähendab platvormi loomist, kus erinevad LED-tabloode kasutajad saaksid enda loodud visuaale, sõnumeid ning animatsioone rakendusesiseselt ja -väliselt jagada, eksportida või importida. Taolise

funktsionaalsuse arendamine nõuaks rakenduse ühendamist mingisuguse andmebaasiga, kuna kõigi kasutajate kujunduste hoiustamine muutuks ühe seadme jaoks liialt mahukaks. Antud edasiarendus tooks rakendusele kindlasti kasutajaid juurde, seega tuleks lisada juurde ka keelevalikuid, kuna praegu on saadaval vaid eesti ja inglise keel.

Seoses kõigi uute funktsionaalsustega võib rakendus minna juba nii suureks, et tuleks kindlasti implementeerida ka ühiktestimine, mis tähendab igale komponendile ja funktsioonile testide kirjutamist. See tagab selle, et rakendus püsib pärast uuendusi töökorras ning vigade tekkimisel on neid lihtsam tuvastada. Lisaks tuleb siis hakata mõtlema rakenduse jõudluse parandamisele ehk vähendada abstraktsioone, valida optimaalsemad andmetüübid ning viia kood võimalikult masinalähedaseks.

Loodud süsteemile tuleks edasiarendusena luua ka parem dokumentatsioon, mis tähendab nii LED-tabloole kui ka mobiilirakendusele korraliku kasutusjuhendi loomist. Kasutusjuhendid võiksid olla digitaalsed ning mugavalt kättesaadavad mobiilirakendusest või GitHub hoidlast. Kasutusjuhendid võiksid lisaks kirjalikule osale sisaldada ka videojuhendeid ja näidisolukordi süsteemi kasutamisest. Lisaks peaksid need seoses avatud lähtekoodiga sisaldama juhiseid, kuidas lähtekoodi täiendada ning lisada sinna oma funktsionaalsusi, kui rakendus ei peaks meelepäraselt toimima.

11 Kokkuvõte

Käesoleva lõputöö eesmärgiks oli luua LED-tabloo ja mobiilirakenduse lahendus, mis võimaldaks nutiseadme abil juhtmevabalt kuvada tablool mitmesuguseid visuaale, sõnumeid ja animatsioone. Loodav mobiilirakendus pidi olema kasutajasõbralik, mitmekeelne ning võimaldama Bluetooth'i kasutamist. Lahenduse tarkvara loodi avatud lähtekoodiga, võimaldamaks kogukonnal tulevikus kaasa aidata funktsionaalsuste arendamisel ning vigade parandamisel. Tarkvaraliste vahenditena kasutati arendusel VSCode'i, PlatformIO'd, Xcode'i, React Native raamistikku ning kolmandate osapoolte teke, mis võimaldasid Bluetooth-ühenduse loomist ja andmevahetust ning LED-tablool sisu kuvamist. Riistvara poolelt kasutati WS2812B LED-tablood ning ESP32-S3 mikrokontrolleriga arendusplaati.

Lõputöö käigus sõnastati esmalt nõuded loodava süsteemi jaoks, uuriti LED-tabloode toimimist, soetati vajalik riistvara ning alustati sardtarkvara ja mobiilirakenduse arendusega. Arendusprotsessi alguses tutvuti Bluetooth Low Energy tehnoloogiaga ja mobiilirakenduse arendusega ning valiti sobivad keskkonnad ning raamistik. Protsessi vältel keskenduti heale kasutajakogemusele, eelistati kaasaegseid kujundusi, kasutati versioonihaldust ning järgiti puhta koodi põhimõtteid. Viimistluse faasis tutvuti tarkvaraga Fusion360, et modelleerida kasutatavale arendusplaadile ilme parandamiseks korpus.

Lõpptulemusena valminud LED-tabloo koos juhtrakendusega vastas lõputöös püstitatud nõuetele ning täitis seatud eesmärgid. Valmis kasutajasõbralik funktsionaalne rakendus, mille abil saab nutiseadme Bluetooth'i abil LED-tablooga ühendada ja tablool kuvatavat sisu valida, ning esialgne versioon ESP32-S3 mikrokontrolleriga ühendatud LED-tabloost, mille juhtimist loodud rakendus toetab. Valminud LED-tablood saab sisetingimustes pärast USB-C abil vooluvõrku ühendamist kasutada sõnumite, visuaalide ning animatsioonide kuvamiseks, kas informatiivsel või meelelahutuslikul eesmärgil. Lõputööl on mitmeid võimalikke edasiarendusi, kuna seatud lisaeesmärgid ei saanud täidetud ning autoril on isiklik motivatsioon edasiseks tootearenduseks.

Kasutatud kirjandus

- [1] A. Pennington, „Screen tech takes a leap forward,“ 23. jaanuar 2024. [Võrgumaterjal]. Available: <https://etech.iec.ch/issue/2024-01/screen-tech-takes-a-leap-forward>. [Kasutatud 10. aprill 2024].
- [2] G. Cocca, „How to Write Clean Code – Tips and Best Practices (Full Handbook),“ 15. mai 2023. [Võrgumaterjal]. Available: <https://www.freecodecamp.org/news/how-to-write-clean-code/>. [Kasutatud 5. aprill 2024].
- [3] tonym0, „Getting Started,“ 31. jaanuar 2024. [Võrgumaterjal]. Available: <https://kno.wled.ge/basics/getting-started/>. [Kasutatud 10. aprill 2024].
- [4] Nordic Semiconductor, „The Difference Between Classic Bluetooth and Bluetooth Low Energy,“ 26. mai 2021. [Võrgumaterjal]. Available: <https://blog.nordicsemi.com/getconnected/the-difference-between-classic-bluetooth-and-bluetooth-low-energy>. [Kasutatud 10. aprill 2024].
- [5] Silicon Laboratories, „[TROUBLESHOOTING]: Unable to discover BLE peripheral in iOS/Android Bluetooth Se,“ 9. juuli 2021. [Võrgumaterjal]. Available: https://community.silabs.com/s/article/x-troubleshooting-unable-to-discover-ble-peripheral-in-ios-android-bluetooth-se?language=en_US. [Kasutatud 10. aprill 2024].
- [6] yfxiong, „CoolLED1248,“ 24 aprill 2024. [Võrgumaterjal]. Available: https://play.google.com/store/apps/details?id=com.jtkj.led1248&hl=en_US&pli=1. [Kasutatud 27 aprill 2024].
- [7] Avago Technologies, „Introduction to Driving LED Matrices,“ 11. juuli 2013. [Võrgumaterjal]. Available: <https://docs.broadcom.com/doc/AV02-3697EN>. [Kasutatud 5. aprill 2024].
- [8] Waveshare Electronics, „RGB-Matrix-P2.5-64x64,“ 10. aprill 2024. [Võrgumaterjal]. Available: <https://www.waveshare.com/wiki/RGB-Matrix-P2.5-64x64>. [Kasutatud 10. aprill 2024].
- [9] Worldsemi, „WS2812B,“ 10. aprill 2024. [Võrgumaterjal]. Available: <https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf>. [Kasutatud 10. aprill 2024].
- [10] PixelFLEX, „LED DIGITAL DISPLAY ESSENTIALS: PIXEL PITCH, MATRIX AND VIEWABLE AREA,“ 10. aprill 2024. [Võrgumaterjal]. Available: <https://pixelflexled.com/led-digital-display-essentials-pixel-pitch-matrix-viewable-area/>. [Kasutatud 10. aprill 2024].
- [11] N. Poole, „Everything You Didn't Want to Know About RGB Matrix Panels,“ 30. märts 2018. [Võrgumaterjal]. Available: <https://www.sparkfun.com/news/2650>. [Kasutatud 10. aprill 2024].
- [12] Micro Robotics, „Wave P3 RGB Full-Color LED Matrix Panel,“ 10. aprill 2024. [Võrgumaterjal]. Available: <https://i.ibb.co/RgYVfV6/RGB-Matrix-P3-64x32-details-7.jpg>. [Kasutatud 10. aprill 2024].

- [13] F. logic, „Dot matrix,“ 27. oktoober 2023. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Dot_matrix. [Kasutatud 10. aprill 2024].
- [14] DOIT VISION, „How to calculate LED screen resolution, size & viewing distance,“ 17. aprill 2024. [Võrgumaterjal]. Available: <https://www.doitvision.com/calculate-led-screen-resolution-size-viewing-distance/>. [Kasutatud 20. aprill 2024].
- [15] K. Cundall, „LED DISPLAYS AS A DIRECT REPLACEMENT FOR LCD VIDEO WALLS - HOW DO THEY MEASURE UP?,“ 16. aprill 2023. [Võrgumaterjal]. Available: <https://www.theledstudio.co.uk/blog/led-as-a-replacement-for-lcd>. [Kasutatud 10. aprill 2024].
- [16] Focus Technology Co., Ltd, „Customized Indoor Flexible LED Crystal Film Screen Full Color Transparent LED Flexible Display,“ 10. aprill 2024. [Võrgumaterjal]. Available: <https://zxbx371.en.made-in-china.com/product/qEvpbeMGCOhn/China-Customized-Indoor-Flexible-LED-Crystal-Film-Screen-Full-Color-Transparent-LED-Flexible-Display.html>. [Kasutatud 10. aprill 2024].
- [17] WhaleCo Inc., „1pc GOTUS LED Matrix Panel Rolling Advertising LED Signage USB 5V Bluetooth App Controlled Sign Light Programmable LED Car Sign,“ 20. aprill 2024. [Võrgumaterjal]. Available: https://www temu.com/ee-en/1pc-led-matrix-panel-rolling-advertising-led-signage-usb-5v--app-controlled-sign-light-programmable-led-car-sign-g-601099543373736.html?_oak_mp_inf=EKjPh6am1ogBGiBlZmNmMWMzMDAwNjI0ZWU1YWVjZDM4ZTRlNGRkNTcyMCC3jbnL9DE%3D&top_gall. [Kasutatud 20. aprill 2024].
- [18] AZ-Delivery, „RGB Panel 8x32 Flex,“ 20. aprill 2024. [Võrgumaterjal]. Available: <https://cdn.shopify.com/s/files/1/1509/1638/files/8x32RGBPanelFlex.pdf?v=1675852016>. [Kasutatud 20. aprill 2024].
- [19] aliexpress.com, „WS2812B LED Digital Flexible Individually Addressable Panel Light WS2812IC 8x8 16x16 Module Matrix Screen DC5V,“ 20. aprill 2024. [Võrgumaterjal]. Available: https://www.aliexpress.com/item/4001298469994.html?spm=a2g0o.detail.pcDetailTopMoreOtherSeller.6.46d3SEBgSEBgwk&gps-id=pcDetailTopMoreOtherSeller&scm=1007.40050.354490.0&scm_id=1007.40050.354490.0&scm-url=1007.40050.354490.0&pvid=dbfd0866-9fa5-4b23-872b-3. [Kasutatud 20. aprill 2024].
- [20] robolabor.ee, „PAINDUV 8X32 RGB LED MAATRIKS,“ 20. aprill 2024. [Võrgumaterjal]. Available: <https://robolabor.ee/et/ekraanid-ja-naidikud/1182-painduv-8x32-rgb-led-maatriks.html>. [Kasutatud 20. aprill 2024].
- [21] Espressif Systems (Shangai) Co., Ltd, „ESP8266EX,“ 1. juuni 2023. [Võrgumaterjal]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Kasutatud 10. aprill 2024].
- [22] Amazon.com, Inc, „2/3 pieces ESP32 development board NodeMCU modules: ESP32 type C NodeMCU development board, ESP-WROOM-32, 2.4GHz dual-mode WiFi + Bluetooth dual cores microcontroller integrated, CH340C chip (2,“ 20. aprill 2024. [Võrgumaterjal]. Available: https://www.amazon.de/-/en/dp/B0C86C3JM5?ref=ppx_yo2ov_dt_b_product_details&th=1. [Kasutatud 20. aprill 2024].
- [23] BOTLAND, „ESP32-S3-DEV-KIT-N8R8 - development board WiFi + Bluetooth - without connectors - Waveshare 24363,“ 10. aprill 2024. [Võrgumaterjal].

- Available: <https://botland.store/esp32-wifi-and-bt-modules/23340-esp32-s3-dev-kit-n8r8-development-board-wifi-bluetooth-without-connectors-waveshare-24363.html>. [Kasutatud 10. aprill 2024].
- [24] Arduino, „Overview of the Arduino IDE 1,“ 17. jaanuar 2024. [Võrgumaterjal]. Available: <https://docs.arduino.cc/software/ide-v1/tutorials/Environment/>. [Kasutatud 10. aprill 2024].
- [25] Microsoft, „Getting Started,“ 11. aprill 2024. [Võrgumaterjal]. Available: <https://code.visualstudio.com/docs>. [Kasutatud 30. aprill 2024].
- [26] I. Kravets, „PlatformIO IDE,“ 26. aprill 2022. [Võrgumaterjal]. Available: <https://docs.platformio.org/en/latest/integration/ide/pioide.html#>. [Kasutatud 10. aprill 2024].
- [27] Apple, „Xcode,“ 10. aprill 2024. [Võrgumaterjal]. Available: <https://developer.apple.com/documentation/xcode>. [Kasutatud 10. aprill 2024].
- [28] H. Akhtar, „iOS Development on Windows: A Complete Guide to Xcode for Windows,“ 19. mai 2023. [Võrgumaterjal]. Available: <https://www.browserstack.com/guide/xcode-for-windows>. [Kasutatud 20. aprill 2024].
- [29] Autodesk Inc., „Autodesk Fusion 360 for personal use,“ 30. aprill 2024. [Võrgumaterjal]. Available: <https://www.autodesk.com/products/fusion-360/personal>. [Kasutatud 30. aprill 2024].
- [30] R. Feranec, „How to Make Custom ESP32 Box in 1 Hour | Full Tutorial,“ 18. aprill 2023. [Võrgumaterjal]. Available: <https://www.youtube.com/watch?v=ZCGhYYpyf84>. [Kasutatud 10. aprill 2024].
- [31] M. contributors, „Web Bluetooth API,“ 11. märts 2024. [Võrgumaterjal]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Web_Bluetooth_API. [Kasutatud 15. aprill 2024].
- [32] atharvaso, „Introduction of Mobile Applications,“ 23. jaanuar 2023. [Võrgumaterjal]. Available: <https://www.geeksforgeeks.org/introduction-of-mobile-applications/>. [Kasutatud 15. aprill 2024].
- [33] Meta Platforms, Inc, „React Native,“ 15. aprill 2024. [Võrgumaterjal]. Available: <https://reactnative.dev/>. [Kasutatud 15. aprill 2024].
- [34] A. Polyakov, „The Six Most Popular Cross-Platform App Development Frameworks,“ 22. aprill 2024. [Võrgumaterjal]. Available: <https://www.jetbrains.com/help/kotlin-multiplatform-dev/cross-platform-frameworks.html#how-do-you-choose-the-right-cross-platform-app-development-framework-for-your-project>. [Kasutatud 25. aprill 2024].
- [35] E. You, „Comparison with Other Frameworks,“ 20. jaanuar 2022. [Võrgumaterjal]. Available: <https://v2.vuejs.org/v2/guide/comparison.html?redirect=true>. [Kasutatud 25. aprill 2024].
- [36] Meta Platforms, Inc, „Setting up the development environment,“ 22. aprill 2024. [Võrgumaterjal]. Available: <https://reactnative.dev/docs/environment-setup>. [Kasutatud 25. aprill 2024].
- [37] E. Stevens, „7 fundamental UX design principles all designers should know,“ 22. juuni 2022. [Võrgumaterjal]. Available: <https://www.uxdesigninstitute.com/blog/ux-design-principles/>. [Kasutatud 10. aprill 2024].
- [38] Expo, „@expo/vector-icons@14.0.0,“ 1. aprill 2024. [Võrgumaterjal]. Available: <https://icons.expo.fyi/Index>. [Kasutatud 10. aprill 2024].

- [39] G. Fonts, „Inter,“ 2. mai 2024. [Võrgumaterjal]. Available: <https://fonts.google.com/specimen/Inter>. [Kasutatud 2. mai 2024].
- [40] D. Friyia, „Dan's React Native Lab,“ 8. aprill 2024. [Võrgumaterjal]. Available: <https://www.youtube.com/@DanRNLab>. [Kasutatud 20. aprill 2024].
- [41] dotintent, „Introduction,“ 1. jaanuar 2024. [Võrgumaterjal]. Available: <https://dotintent.github.io/react-native-ble-plx/>. [Kasutatud 20. aprill 2024].
- [42] i18next, „Introduction,“ 1. jaanuar 2024. [Võrgumaterjal]. Available: <https://www.i18next.com/>. [Kasutatud 20. aprill 2024].
- [43] Redux, „Getting Started with Redux,“ 31. märts 2024. [Võrgumaterjal]. Available: <https://redux.js.org/introduction/getting-started>. [Kasutatud 1. mai 2024].
- [44] C. Kalika, „Redux Persist,“ 1. juuni 2021. [Võrgumaterjal]. Available: <https://www.npmjs.com/package/redux-persist>. [Kasutatud 1. mai 2024].
- [45] P. Burgess, K. Rembor ja I. ada, „Best Practices,“ 8. märts 2024. [Võrgumaterjal]. Available: <https://learn.adafruit.com/adafruit-neopixel-uberguide/best-practices>. [Kasutatud 25. aprill 2024].
- [46] Espressif Systems (Shanghai) Co., Ltd, „ESP32-S3-WROOM1 Datasheet,“ 24. november 2023. [Võrgumaterjal]. Available: https://www.espressif.com/sites/default/files/documentation/esp32-s3-wroom-1_wroom-1u_datasheet_en.pdf. [Kasutatud 30. aprill 2024].
- [47] K. Ren, „Higher Speed How Fast Can It Be?,“ 20. veebruar 2017. [Võrgumaterjal]. Available: <https://www.bluetooth.com/blog/exploring-bluetooth-5-how-fast-can-it-be/>. [Kasutatud 1. mai 2024].
- [48] Bluetooth SIG, „The Bluetooth® Low Energy Primer,“ 15. märts 2024. [Võrgumaterjal]. Available: <https://www.bluetooth.com/wp-content/uploads/2022/05/the-bluetooth-le-primer-v1.2.0.pdf>. [Kasutatud 1. mai 2024].
- [49] RandomNerdTutorials.com, „Getting Started with ESP32 Bluetooth Low Energy (BLE) on Arduino IDE,“ 16. mai 2019. [Võrgumaterjal]. Available: <https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/>. [Kasutatud 5. mai 2024].
- [50] Expo, „Introduction to Expo Router,“ 10. aprill 2024. [Võrgumaterjal]. Available: <https://docs.expo.dev/router/introduction/>. [Kasutatud 30. aprill 2024].
- [51] Expo, „Drawer,“ 1. mai 2024. [Võrgumaterjal]. Available: <https://docs.expo.dev/router/advanced/drawer/>. [Kasutatud 1. mai 2024].
- [52] A. Alabsi, „Reanimated Color Picker,“ 3. märts 2024. [Võrgumaterjal]. Available: <https://alabsi91.github.io/reanimated-color-picker/>. [Kasutatud 5. mai 2024].
- [53] M. Meloche, „Learn How To Use Screens and Navigation in React-Native,“ 21. oktoober 2021. [Võrgumaterjal]. Available: <https://develpreneur.com/learn-how-to-use-screens-and-navigation-in-react-native/>. [Kasutatud 10. aprill 2024].
- [54] A. A. Demarest, „What is Canva? A guide to the graphic design platform's features and capabilities,“ 18. september 2020. [Võrgumaterjal]. Available: <https://www.businessinsider.com/guides/tech/what-is-canva>. [Kasutatud 10. aprill 2024].
- [55] Google, „Buttons: floating action button,“ 20. aprill 2024. [Võrgumaterjal]. Available: <https://m2.material.io/components/buttons-floating-action-button>. [Kasutatud 20. aprill 2024].

- [56] G. Mechling, „android stuff & co.,“ 15. jaanuar 2018. [Võrgumaterjal]. Available: <https://nilhcm.com/iot/reverse-engineering-bluetooth-led-name-badge>. [Kasutatud 1. märts 2024].
- [57] K. Ren, „A New Way to Debug iOS,“ 15. oktoober 2019. [Võrgumaterjal]. Available: <https://www.bluetooth.com/blog/a-new-way-to-debug-iosbluetooth-applications/>. [Kasutatud 5. märts 2024].
- [58] U. Gnadenteich, „Postimees,“ 19. detsember 2012. [Võrgumaterjal]. Available: <https://www.postimees.ee/1079736/reaalajas-tallinna-busside-liikumist-naitavad-infotablood-hakkasid-toole>. [Kasutatud 10. aprill 2024].
- [59] Megameedia Grupp OÜ, „LED,“ 10. aprill 2024. [Võrgumaterjal]. Available: <https://www.megameediagrupp.ee/teenused/ledkraanide-muuk/led/>. [Kasutatud 10. aprill 2024].
- [60] Transpordiamet, „Muutuva teabega liiklusmärgid,“ 29. jaanuar 2024. [Võrgumaterjal]. Available: <https://www.transpordiamet.ee/muutuva-teabega-liiklusmärgid>. [Kasutatud 10. aprill 2024].
- [61] Kecko, „Bad Ragaz SBB - Bus Stop,“ 8. august 2021. [Võrgumaterjal]. Available: <https://www.flickr.com/photos/kecko/51366052710>. [Kasutatud 5. aprill 2024].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

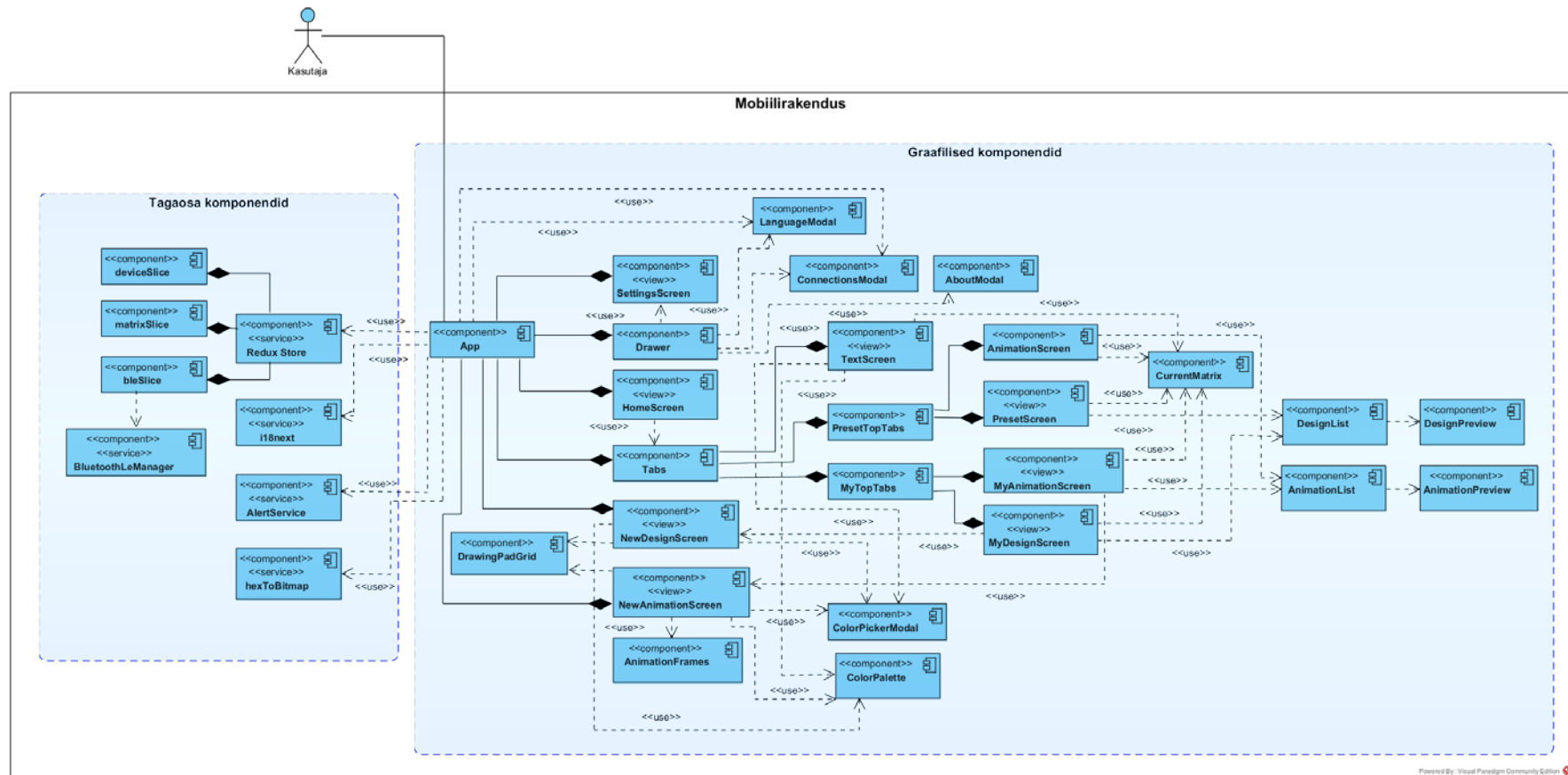
Mina, Rannar Randmäe

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose “LED-tabloo täiendamine juhtmevabaks ühenduseks nutiseadmega“, mille juhendaja on Peeter Ellervee
 - 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.

10.05.2024

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

Lisa 3 – Mobiilirakenduse komponentdiagramm



Lisa 4 – Lõpplahenduse elektriskeem

