

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Krisseliine Pärt**  
**Tark toidukaal**  
**Bakalaureusetöö (9 EAP)**

Juhendajad: Alo Peets  
Anne Villems  
Taavi Duvin

Tartu 2017

## **Tark toidukaal**

### **Lühikokkuvõte:**

Käesoleva bakalaureusetöö raames valmib mobiilirakendus iOS operatsioonisüsteemiga seadmele, mis ühildub sinihamba toidukaaluga. Rakenduse peamiseks eesmärgiks on Eestis elava inimese toitumise jälgimise protsessi lihtsustamine. Esimeses peatükis selgitatakse tasakaalustatud toitumise olemust ning vajalikkust. Samuti analüüsitakse Eestis elavate inimeste toitumisharjumusi ning kaaluprobleeme. Teises peatükis kirjeldatakse sinihamba toidukaalu valmistamist ning ühildamist seadmega. Lisaks kaetakse teises peatükis tarkvaraarendusega seotud etapid.

### **Võtmesõnad:**

Toidukaal, rakendus, bluetooth, Arduino, tasakaalustatud toitumine

**CERCS:** P170 (Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine)

## **Smart Scale**

### **Abstract:**

The purpose of this bachelor thesis is to develop a mobile application for iOS operating system that connects with Bluetooth food scale. The main purpose of this application is to simplify the process of monitoring the diet for people who live in Estonia. The first chapter introduces the nature and necessity of a balance diet. It also analyses the eating habits and weight problems of the people living in Estonia. The second chapter describes how to make and connect with device Bluetooth food scale. In addition, the second chapter covers the software development phases.

### **Keywords:**

Food scale, application, Bluetooth, Arduino, balanced diet

**CERCS:** P170 (Computer science, numerical analysis, systems, control)

## Sisukord

Sissejuhatus .....	5
1. Tasakaalustatud toitumine .....	7
1.1 Toitumise head tavad .....	7
1.2 Inimeste kaaluprobleemid Eestis .....	8
1.3 Toitumine Eestis .....	10
1.4 Eestlaste toitumise jälgimine .....	11
2. Sinihamba toidukaal .....	12
2.1 Toidukaalu komponendid .....	12
Iteduino UNO arendusplaat .....	12
Kaalumõõtmise element ehk andur koos korpusega .....	13
Analoog-digitaalmuundur .....	13
Sinihamba madala energia laiendusplaat .....	13
Muud komponendid .....	15
Isevalmistatud toidukaalu maksumus .....	15
2.2 Toidukaalu valmistamine .....	16
Arduino UNO arendusplaadi programmeerimine .....	16
Madala sinihamba energia laiendusplaadi ühendamise arendusplaadiga .....	18
Kaalunäidu saatmine Arduino UNO arendusplaati .....	20
Arendusplaadile laetav lõplik kood .....	24
3. Targa toidukaalu mobiilirakendus .....	26
3.1 Visioon .....	26
3.2 Kasutatavad tehnoloogiad .....	26
3.3 Olemasolevate toodete kirjeldus .....	26
3.4 Nõuete Analüüs .....	27
Funktsionaalsed nõuded .....	27
Mittefunktsionaalsed nõuded .....	28
3.5 Sinihamba kaalunäidu kuvamine rakenduses .....	29
3.6 Mobiilirakenduse kasutajaliides .....	34
Põhivaade .....	34
Otsinguvaade .....	34
Päeviku vaade .....	35
3.7 Toidud rakenduses .....	36
3.8 Kalendri kuvamine .....	37
3.9 Lahenduse failide arhitektuur .....	38

3.10	Tulevikuplaani võimalused .....	38
3.11	Valminud lahendus.....	39
4.	Kokkuvõte .....	42
5.	Viidatud kirjandus .....	44
Lisad.....		46
I.	Litsents .....	46

## Sissejuhatus

Tervise Arengu Instituudi koostatud uuringust selgus, et 51,7% inimestest on ülekaalulised ning 2,5% on alakaalulised [1]. Uuringus kasutati kaalu hindamiseks indeksit, mis saadakse inimese kehakaalu (kilogrammides) jagamisel pikkuse (meetrites) ruuduga. Kaaluindeks ei kirjelda aga kõige paremini inimese tegelikku kehalist seisundit, sest ei arvestata eraldi lihas-, rasva- ning luumassi. Nimetatud kogukaalu komponentide eraldi arvestamine annaks oluliselt objektiivsema tulemuse. Käesoleva töös on eeldatud, et uuringu tulemused erinevad tegelikkusest vaid mõned protsendid, kuna kaaluindeks annab väärade tulemuste vaid inimeste puhul, kellel on kas väga suur lihassmass ja/või luumass ning väike rasvamass. Need inimesed moodustavad aga rahvastikust väikese osa. Seega on umbes 54,4% eestlasi, kellel on kaaluprobleemid ning kes peaksid normaalkaalu saavutamiseks jälgima toitumist või suurendama kehalist aktiivsust.

Toitumist peaksid tegelikult jälgima kõik inimesed, kuna õigesti ehk tasakaalustatult toitumine on hea tervise aluseks. Paljud normaalkaalulised inimesed soovivad samuti enda toitumist jälgida. Toitumise jälgimise eesmärgiks võib olla näiteks kaalu säilitamine või võistluseks valmistumine.

Toitumise jälgimiseks on erinevaid viise. Eestlaste seas on populaarsed näiteks MyFitnessPal ning Nutridata rakendused. Nimetatud rakendustes on võimalik käsitsi sisestada toidu kaal ning seejärel valida millise toiduga on tegu. Seejärel arvutab rakendus vastavalt sisestatud kaalule ning toidule toitainete kogused ning energiaväärtuse ja lisab toidu koos toitumisalase informatsiooniga kalendrisse. Rakendused võimaldavad kasutajatel saada kiire ülevaate söödud toidu toiteväärtusest. Autori hüpotees ja katsed näitavad, et kumbki neist rakendustest ei ühildu ühegi toidukaaluga. Leidub rakendusi, mis ühilduvad toidukaaluga, kuid mis ei ole mõeldud Eestis kasutamiseks. Ükski toidukaaluga ühilduv rakendus ei ole eestikeelne ning nende tootekataloogis puuduvad paljud Eestis müüdivad toidud.

Kuivõrd praegu pole ühtegi Eestis kasutamiseks loodud rakendust, mis ühilduks toidukaaluga, siis käesoleva töö raames valmistatakse sinihamba toidukaaluga ühilduv toitumise jälgimise mobiilirakendus. Ning sobilik kaal ehitatakse samuti töö raames.

Seega käesoleva töö eesmärkideks on:

1. Selgitada mis on tasakaalustatud toitumine ning miks on see oluline;
2. Kirjeldada Eestis elavate inimeste toitumisharjumusi ning kaaluprobleeme;
3. Uurida olemasolevaid targa toidukaalu rakendusi;
4. Nõuetele vastava sinihamba toidukaalu valmistamine;
5. Valmistada rakendus, kus saab Eestis elav inimene toitumist jälgida ning mis ühilduks sinihamba toidukaaluga;

Töö esimeses peatükis selgitatakse tasakaalustatud toitumise põhitõdesid ning uuritakse Eestis elavate inimeste toitumisharjumusi ning kaaluprobleeme. Teises peatükis kirjeldatakse sobiva sinihamba toidukaalu leidmise ning valmistamise protsessi. Kolmandas peatükis kirjeldatakse mobiilirakenduse arendamise protsessi.

## 1. Tasakaalustatud toitumine

Tasakaalustatud toitumisega saab inimese organism kätte vajalikud toitained, mida on vaja organismi korralikuks funktsioneerimiseks. Targa toidukaalu mobiilirakendus aitab kasutajal tasakaalustatud toituda, arvutades vastavalt kaalunäidule toidu makrotoitainete ning kalorite kogused. Päevase kilokalorsuse ja toitainete jaotuse jaoks on vaja teada mõningaid tervisliku toitumise põhitõdesid, et arvutada võimalikult täpne tulemus iga inimese jaoks. Järgnevalt kirjeldatakse kuidas toituda tasakaalustatult ning analüüsitakse eestlaste kaaluprobleeme, toitumist ning uuritakse, mida võiksid eestlased paremini teha.

### 1.1 Toitumise head tavad

Järgnevad tasakaalustatud toitumise soovitused on mõeldud Eestis elavale täiskasvanud inimesele, tõsisemaid haigusi põdevad või lapseootel inimesed peaksid järgima arsti või professionaalse nõustaja soovitusi. Üldiselt on toitumissoovitused riigiti erinevad keskkonna ning bioloogiliste erisuste pärast ning seega on soovituslik järgida enda elamispiirkonnale koostatud toitumissoovitusi.

Toidust saadud energia peab päevas ära katma organismi põhiainevahetuse-, toidu seedimise, omastamise ning kehalise aktiivsuse energiakulu, nii säilitatakse kaalu [2]. Kaalu langetamiseks peaks päevast kalorite tarbimist vähendada ning tõstmiseks suurendama. Üldiselt arvestatakse päevase kalorsuse arvutamisel kehakaalu, pikkuse, vanuse, soo ja inimese aktiivsuse parameetritega. Meeste ja naiste põhiainevahetuse kalorsuse arvutamiseks kasutatakse Harris-Benedict'i valemit [3]. Enamuse energiast saab organism toiduga. Valem annab tulemuse kilokalorites, mis on levinuim toiduenergia mõõtmise ühik. Põhiainevahetuse kalorsuse  $P_m$  valem avaldub kujul  $66,5 + (13,75 \times \text{kaal kilogrammides}) + (5,003 \times \text{pikkus sentimeetrites}) - (6,755 \times \text{vanus aastates})$  ning naiste põhiainevahetuse kalorsuse  $P_n$  valem avaldub kujul  $655,1 + (9,563 \times \text{kaal kilogrammides}) + (1,850 \times \text{pikkus sentimeetrites}) - (4,676 \times \text{vanus aastates})$ . Põhiainevahetuse kalorsus tuleb veel kordajaga läbi korrutada, et saada kätte inimese päevane kalorsus ning kordaja valitakse vastavalt inimese kehalise aktiivsusele, kordajad on koos kirjeldusega näha tabelis Tabel 1. Näiteks naise, kes on 30-aastane, 170 cm pikk, kaalub 65 kg, päevane kalorsus keskmist natukene kõrgema aktiivsusega arvutatakse vastavalt  $1,9 \times P_n = 1,9 \times (655,1 + (9,563 \times 65) + (1,850 \times 170) - (4,676 \times 30)) = 1,9 \times 1450,915 = 2756,7385 \approx 2757$  kilokalorit.

Tabel 1. Kehalise aktiivsuse tasemest tulenev suhe põhiainevahetuse energiakulusse (PAL) arvestades töö ja vaba aja kehalist aktiivsust [2]

Kehaline aktiivsus	Kordaja
Väga vähene kehaline koormus. Istuv töö, mis ei sisalda liikumist ning väga vähene või puuduv vaba aja kehaline aktiivsus	PAL 1,4–1,5
Vähene kehaline koormus. Istuv töö koos mõningase vajadusega liikuda ja vähene vaba aja kehaline aktiivsus	PAL 1,6–1,7
Keskmine kehaline koormus. Töö, mis sisaldab nii seismist kui ringi liikumist	PAL 1,8–1,9
Kõrge kehaline koormus. Suurt kehalist aktiivsust nõudev töö või igapäevane võistlusspordiks vajalik kehaline treening	PAL 2,0–2,2

Tasakaalustatud toitumise jaoks on vaja, et päevane toitumine oleks võimalikult mitmekesine. Selleks on vaja jälgida toitainete ehk valkude, rasvade, süsivesikute, mineraalainete ja vee tarbimist. Toitainetel on alanimetus makrotoitained kuhu kuuluvad valgud, rasvad ning süsivesikud. Rahvakeeli jagatakse kõik makrotoitained veel oma korda kaheks headeks ja halbadeks, seega tasakaalustatud toitumise jaoks peaks tarbima organismile kasulikke ehk häid toiduaineid. Üldiselt on organismile kasulikud toiduained need, mida ei ole liigselt töödeldud näiteks juur-, köögi- ja puuviljad, liha, kala ning muna. Tervisele kahjulikud toidud sisaldavad rohkelt küllastunud rasvhappeid, suhkruid ja/või soola, neid peaks tarbima vähem ja harvemini [4]. Optimaalselt võiks valke olla 10-15%, ja rasvu 25-30% päevasest toiduenergiast ning ülejäänud süsivesikud [2].

Tihti peale inimesed ei jälgi toitainete osakaalu ja söödud kilokalorite hulka päevas, mis võivad põhjustada erinevaid probleeme nagu ülekaalulisus, alakaalulisus ja haigusi. Järgmises peatükis tutvustatakse 2014. a läbiviidud eestlaste toitumisharjumuste uuringut.

## 1.2 Inimeste kaaluprobleemid Eestis

Kaaluprobleemid võivad tekkida kui ei sööda tasakaalustatult, sest siis ei saa inimese organism vajalikke toitaineid kätte ja seega keha sunnib inimest rohkem toitu tarbima [4].

Tervise Arengu Instituudi koostatud uuringus kasutatakse kaalu hindamiseks kaaluindeksit, mis saadakse kehakaalu kilogrammides jagamisel pikkuse meetrites ruudus [1]. Kaaluindeks ei ole kõige parem hindamaks ülekaalulisust, sest indeks ei arvesta lihasmassi ja rasvamassi suhet ning luumassi. Näiteks inimene, kellel on rasvamass küllaltki madal ent lihasmass tavalisest oluliselt kõrgem, on kaaluindeksi järgi ülekaaluline. Käesoleva töö raames saab siiski eeldada, et neid inimesi, keda ei kirjelda kaaluindeks õigesti, on piisavalt vähe, ning see ei mõjuta oluliselt tulemust. Kaaluindeksi ehk KI hindamisel võeti aluseks WHO poolt soovitatud skaalat, kus KI alla 18,5 tähistab alakaalu, vahemik 18,5-24,9 normaalkaalu, vahemik 25-29,9 ülekaalu ning üle 30 tähistab rasvumist. Uuringust selgus, et Eestis on üle 18-aastaseid ülekaalulisi ja rasvunud inimesi 51,7% kogu nende arvust [1]. Ning alakaalulisi täiskasvanuid on 2,5% kogu nende arvust. Täiskasvanud inimesed suudavad juba enda toidulauda paremini kontrollida, kuid ilmselt jääb puudu teadmistest või tervisliku toitumise jälgimise mugavatest vahenditest, et toituda tasakaalustatult. Eesti ülekaaluliste protsent on suurem Euroopa Liidu keskmisest, mis on meestel 16,1% ja naistel 15,7% [5].

Eestis võib olla ülekaalu tekke põhjuseks liigne süsivesikute tarbimine, alkoholi liigtarbimine, üle söömine ja vähene liikumine, harvem haigused. Liigne süsivesikute tarbimine võib tulla sellest, et süsivesikurikkad toidud on tihti peale odavamad võrreldes tervislikema alternatiividega ja kergesti kätte saadavad nagu näiteks kondiitritooted, krõpsud, kartul ja makaronid. Liigne söömine võib olla tingitud stressist, igavusest, toidukordade vähesusest või vee tarbimisest. 2015. aasta lõikes tarbisid üle 15-aastased eestlased keskmiselt 10,3 liitrit alkoholi inimese kohta, mida on ilmselgelt liiga palju [6]. Alkoholi ei ole inimese organismile vajalik ja seega on need kehale tühjad kalorigid. Tühjaks kaloriks (ingl.k. empty calories) nimetatakse selliseid toiduaineid, milles ei ole inimesele kasulikke aineid ning mis mõjuvad halvasti tervisele [7].

Eestis ei ole alakaal nii suur probleem kui ülekaalulisus, kuid mõlemad probleemid on sama tähtsad. Alakaalu põhjuseks on suuresti liiga vähese toiduenergia tarbimine ning harvem haigused. Paljud inimesed arvavad, et süües vähe ja tehes väga suure koormusega trenni aitab see neil saavutada soovitud tulemust. Tegelikult sellise eluviisiga kulutatakse väga palju lihast ning nõrgendatakse organismi immuunsussüsteemi, mistõttu on igasugused haigused kiirelt tulema.

Peamiseks kaaluprobleemiks on Eestis ülekaalulisus, kuid alakaalu tuleb käsitleda sama tõsise terviseprobleemina. Probleemi lahendamiseks peaksid inimesed analüüsima enda toitumist ning tulemuste saamiseks ka toitumist jälgima.

### **1.3 Toitumine Eestis**

Eestis viidi aastal 2016 läbi uuring, kus küsiti eestlastelt nende toitumisharjumiste kohta [1]. Selle uuringu põhjal on võimalik üldiselt kirjeldada eestlaste toitumisharjumisi.

Hommikusöök on väga tähtis osa tasakaalustatud toitumisest, nimelt on selgunud, et inimesed kes söövad hommikusööki, näksivad päeva jooksul toidukordade ajal vähem ning seega suudavad hoida enda kehakaalu kontrolli all [8]. Uuringust selgus, et vastajatest 79,7% söövad enamasti hommikusööki, mis on väga positiivne tulemus [1].

Kõik inimesed peaksid tarbima iga päev puu- ja köögivilju. Ainult 26,1% uuringul osalejatest sööb värskeid köögivilju kuuel kuni seitsmel päeval nädalas ning 31,9% vastanutest tarbib puuvilju ja marjasid kuuel või kõikidel päevadel nädalas [1]. Puu- ja köögiviljadest saab inimene kätte vajalikud süsivesikud, vitamiinid ja mineraalid, mis on organismile väga kasulikud.

Ilmselt on igale inimesele teada, et valge jahu on organismile kahjulik, sest seda on töödeldud nii, et inimesele vajalikud toitained on sealt ära kadunud. Uuringul osalejaid, kes tarbivad vähemalt ühe saia viilu päevas, on 30,4% [1]. Inimese organismile ei ole valge sai vajalik seega pigem võiks tarbida alternatiive. Eestis on poodides väga palju erinevate tootjate tooteid, mis asendavad valgelt saia.

Kommid ja šokolaadid ei ole inimesele vajalikud, kui vahel harva võib neid vähestes kogustes tarbida. Uuringus selgus, et 41,9% vastanutest tarbib maiustusi kolmel kuni seitsmel päeval nädalas, mis on liiga tihe tarbimine. Uuringust osalejatest 47,4% tarbib šokolaadi ja komme ühel kuni kahel päeval nädalas, see on küllaltki mõistlik päevade arv kui tarbitakse korraga vähe maiustusi ehk umbes 50 grammi. Organismile oleks kõige parem kui maiustusi tarbitakse maksimaalselt kord kuus ning siis ka mõistlikus koguses.

Uuringu põhjal võib teha järeldusi, et liiga vähe tarbitakse vajalikke puu- ja köögivilju ning maiustusi ja valge jahust saia võiks tarbida vähem. Tasakaalustatud toitumiseks peaks inimene sööma rohkem aedvilju, et organism saaks kätte vajalikke vitamiine ning kasulikke toiduaineid. Vähem peaks tarbima maiustusi ning valget saia, kuna sealsed toitained ei ole organismile vajalikud. Maiustuse ning valge saia vähemaks tarbimiseks ning puu- ja

köögiviljade rohkemaks tarbimiseks oleks hea pidada päevikut, kus kirjutatakse ülesse kõik söödud toidud. Päeviku pidamine võib aidata kaasa paremale toitumisele.

#### **1.4 Eestlaste toitumise jälgimine**

Osad eestlased, kelle eesmärgiks on kaalu langetamine, tõstmine või hoidmine, soovivad täpselt järke pidada enda tarbitud toidust päeva jooksul.

Toitumise jälgimise lihtsustamiseks kasutatakse Eestis peamiselt Myfitnesspal või Nutridata rakendusi. Nendes rakendustes saab inimene sisestada toiduaine kaalu ning nimetuse ja lisada selle enda päevasesse toitumiskalendrisse. Myfitnesspal ja Nutridata teevad ülejäänud töö ehk arvutavad ise toitainete osakaalu ning loevad kokku ka kilokalorid. Eelnevalt mainitud rakendused ei ühildu lõputöös läbiviidud analüüsi kohaselt ühegi targatoidukaaluga, mis võimaldaks mugavamalt koguseid sisestada. Kasutaja peab rakendustes käsitsi sisestama kaalunäidu, mis võib olla tülikas ning võib tekitada vigu, näiteks kui kasutaja unustab sisestada näidu või sisestab ebakorrekse näidu. Samuti ei pruugi kasutajad osata hinnata korrektselt koguseid ning sisestavad vale koguse ja saavad seega väärade tulemuste. Mobiilirakendus ühilduva sinihamba toidukaaluga lahendaks aga need probleemid ning vähendaks käsitsi sisestava informatsiooni hulka päevase toitumise jälgimise protsessis.

## 2. Sinihamba toidukaal

Rakenduse valmistamisel on vajalik kaalunäitu kätte saada, mis saadakse sinihamba tehnoloogial põhinevalt toidukaalult. Müügil on üksikuid sinihamba toidukaalusid, kuid turul müüdavatel puudub API ehk rakenduseliides, mis võimaldaks kätte saada kaalu näitu või avatud lähtekood, mille abil saaks toidukaalu rakendust programmeerida. Muudel juhtmevabadel tehnoloogiatel põhinevaid toidukaalusid on kas väga vähe või ei eksisteeri neid üldse. Sobivate toidukaalude puudumise tõttu oli vajalik lõputöö raames valmistada lisaks tarkvarale ka rakenduse funktsionaalsuste demonstreerimiseks vajalike näitajatega toidukaal. Sinihamba toidukaalu ehitamiseks iOS seadmete ühilduvuse jaoks on vaja erinevaid komponente. Need on vaja omavahel ühendada ning seadistada, et toidukaal töötaks.

### 2.1 Toidukaalu komponendid

Järgnevalt on kirjeldatud toidukaalu ehitamiseks kasutatud komponente ning nende maksumust.

#### Iteaduino UNO arendusplaat

Iteaduino UNO arendusplaat on nähtav joonisel (vt. Joonis 1).



Joonis 1. Iteaduino UNO arendusplaat[9]

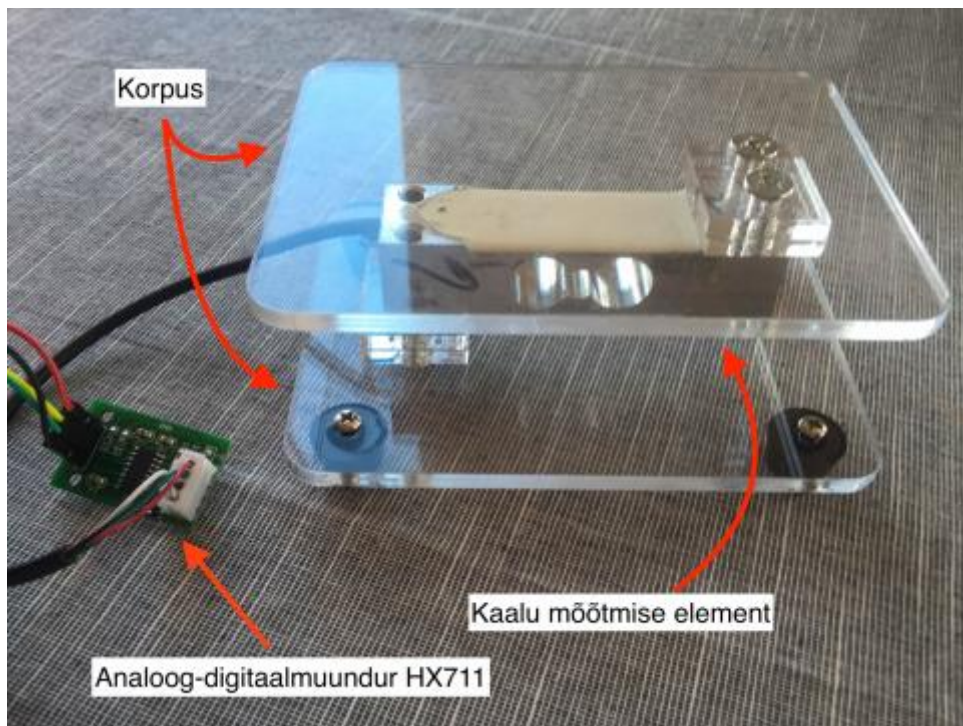
Arendusplaadid suudavad lugeda sisendeid ja saata andmeid [10]. Andmete lugemiseks ja saatmiseks on vaja plaadile laadida koodi ning seejärel seda käivitada. Antud töös saab arendusplaat sisendiks kaalunäidu ning saadab selle sinihamba signaaliga mobiilirakendusse.

### **Kaalumõõtmise element ehk andur koos korpusega**

Kaalumõõtmise element koos korpusega on nähtav joonisel (vt. Joonis 2). Kaalumõõtmise elementi on vaja selleks, et teada mõne objekti kaalu. Korpus muudab kaalumise lihtsamaks, kuna objekti saab panna lihtsalt korpuse peale.

### **Analoog-digitaalmuundur**

Kaalu valmistamisel kasutati HX711 analoog-digitaalmuundurit, mis on nähtav joonisel (vt. Joonis 2).

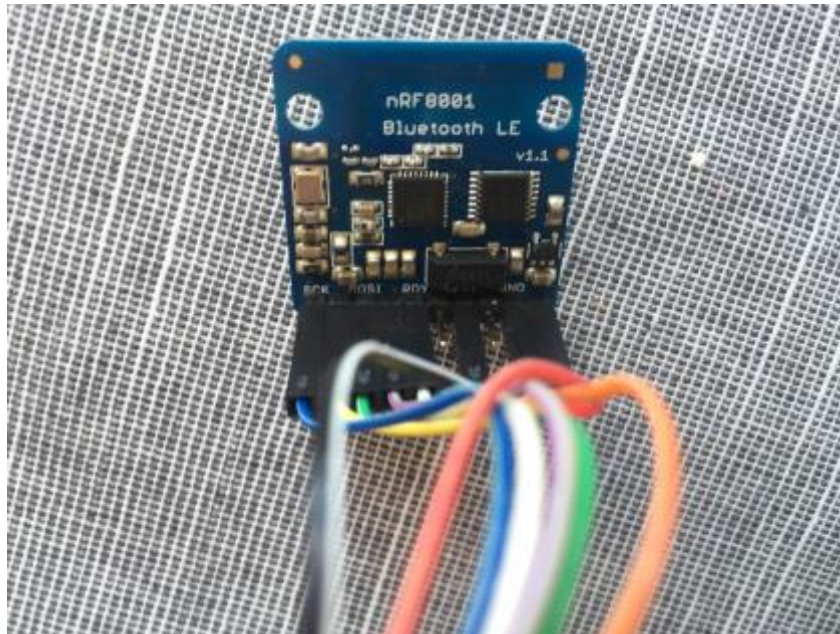


Joonis 2. Kaalu mõõtmise element koos korpuse ja analoog-digitaal muunduriga

Mainitud komponent võimaldab piisava täpsusega lugeda kaalu näitu kaalumõõtmise andurilt.

### **Sinihamba madala energia laiendusplaat**

Käesolevas töös kasutatakse sinihamba Adafruit madala energia laiendusplaati nrF001, mis on nähtav joonisel (vt. Joonis 3).



Joonis 3. Sinihamba madala energia Adafruit laiendusplaat

Laiendusplaadid ühenduvad arendusplaatidega ning on mõeldud erinevate funktsioonide lisamiseks projektile. Sinihamba signaal saadab kaalunäitu seadmetele. Kindlasti peab sinihammas olema madala energiaga. Eestis üks levinumaid sinihamba laiendusplaat (vt. Joonis 4) uuemate iOS seadmetega ei ühildu. Autor proovis erinevaid võimalusi, et saada sinihamba laiendusplaadi signaali kätte iOS operatsioonisüsteemi telefoniga, kuid tulutult kuna uuemad iOS seadmed ei toeta vanema tüüpi sinihamba signaali.

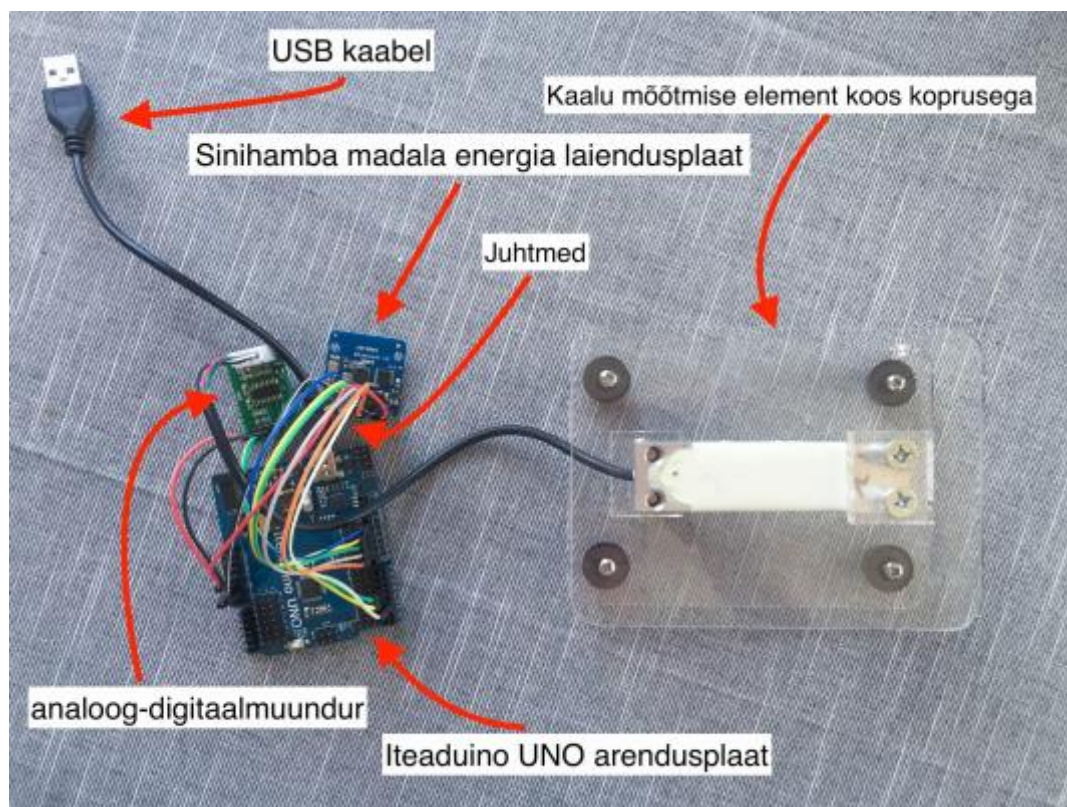


Joonis 4. Sinihamba laiendusplaat

Sinihamba madala energia laiendusplaadi signaali leiavad üles iPhone 5 ja iPad neljanda generatsiooni ning uuemad versioonid.

### **Muud komponendid**

Arendusplaadi programmeerimiseks ning voolu andmiseks on vaja USB kaablit (vt. Joonis 5). Komponentid ühendatakse omavahel kasutades spetsiaalseid juhtmeid (vt. Joonis 5).



Joonis 5. Toidukaal

Toidukaalu valmistamiseks valiti võimalikult odavad ent kvaliteetsed komponendid, et kogu kulud oleksid võimalikult madalad.

### **Isevalmistatud toidukaalu maksumus**

Kogu toidukaalu maksumus kui kasutada valmistamisel eelmises peatükis mainitud komponente on 42,44 eurot. Komponentide maksumus on välja toodud tabelis Tabel 2.

Tabel 2. Toidukaalu komponentide maksumus

Toode	Hind
Iteaduino UNO arendusplaat	5,82 eurot
Kaalu mõõtmise element ehk andur koos korpuse ning HX711 analoog-digitaalmuunduriga	12,72 eurot
Sinihamba madala energia laiendusplaat nRF001	19,95 eurot
USB kaabel	2 eurot
Juhtmed	1,95 eurot
<b>Kokku</b>	<b>42,44 eurot</b>

Arendusplaati on võimalik osta Itead veebipoest (<https://www.itead.cc/>), kaalu mõõtmise elementi koos korpuse ja analoog-digitaalmuunduriga leiab eBayst (<http://www.ebay.co.uk/>). Laiendusplaat ja juhtmed osteti Tallinna YEInternational AS kauplusest ning USB kaablit müüakse tehnikakauplustes.

Järgnevalt on vaja kaalu ehitamiseks komponendid ühendada ning arendusplaat programmeerida.

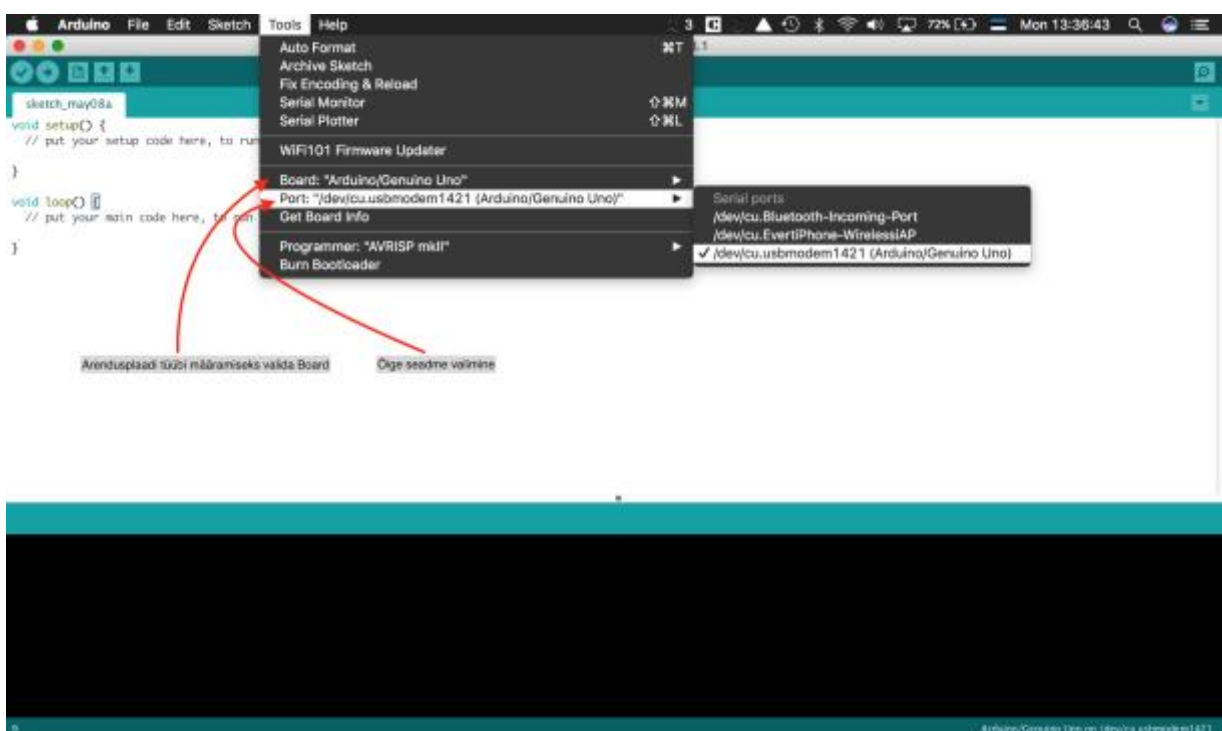
## 2.2 Toidukaalu valmistamine

Toidukaalu valmistamise etapid on kirjeldatud järgnevates alapeatükkides.

### Arduino UNO arendusplaadi programmeerimine

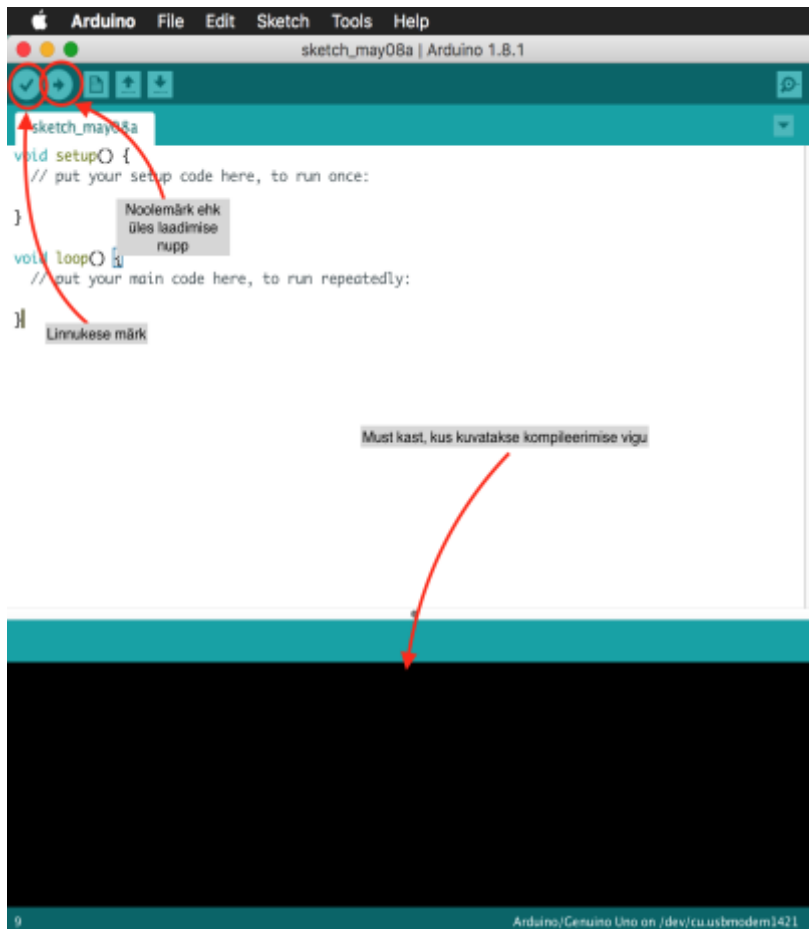
Arendusplaadi programmeerimisel kasutatakse programmeerimiskeelt Arduino ning integreeritud programmeerimiskeskonda Arduino. Arenduskeskkonnas saab koodi kirjutada ning kompileerida. Arenduskeskkonna saab alla laadida Arduino ametlikult kodulehelt <https://www.arduino.cc/en/Main/Software>. MacOS operatsiooni süsteemiga arvutil alla laadimisel on vaja järgida ette antud juhiseid ning pärast alla laadimist on võimalik programm avada. Programmi saab avada näiteks kui vajutada klaviatuuril korraka

käsu (ingl.k. command ehk cmd) ja tühiku (ingl.k. space) nuppu ning avanevasse otsingumootoris kirjutada Arduino ning seejärel vajutada sisestusklahvi (ingl.k. enter) nuppu. Kui programm on avanenud siis kuvatakse automaatselt fail, kuhu on võimalik koodi kirjutada. Seejärel peab ära määrama arendusplaadi tüübi, selleks peab ülevalt menüüs valima *Tools*, seejärel *Board* ning siis peab valima *Arduino/Genuino UNO* (vt. Joonis 6). Nüüd tuleb ühendada arendusplaat arvutiga, selleks peab USB kaabli väiksema otsa ühendama arendusplaadiga ning suurema arvutiga. Seejärel valitakse arenduskeskkonna ülevalt menüüst *Tools* ning seejärel *Port*. Avanevast listist valitakse õige seade (vt. Joonis 6). Kui ei teata milline on õige seade, siis jäetakse meelde nimekirjas kuvatud nimed ning ühendatakse arendusplaat lahti arvutist. See nimi mis ära kaob ongi soovitud arendusplaat. Taas ühendamisel valitakse eelnevalt kadunud nimi.



Joonis 6. Arduino arenduskeskkonnas arendusplaadi tüübi määramine ning õige seadme valimine

Pärast koodi lisamist on võimalik koodi kompileerida. Vigade avastamiseks peab vajutama üleval linnukese märgi peale (vt. Joonis 7). Kui kood ei kompileeri, siis mustas kastis (vt. Joonis 7) kuvatakse punase kirjaga veateade. Koodi laadimiseks plaadile tuleb vajutada noolemärgile ehk üles laadimise nupule (ingl.k. upload) (vt. Joonis 7).



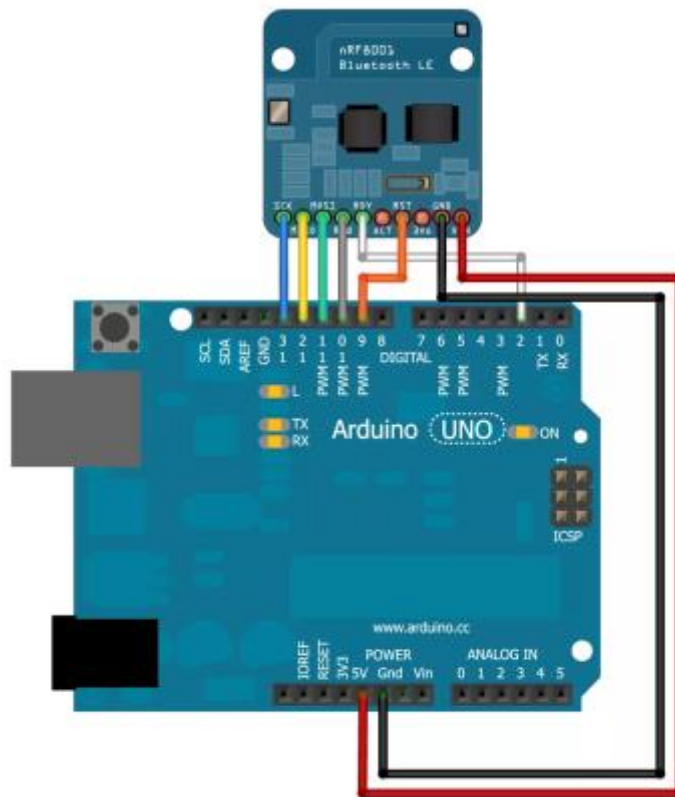
Joonis 7. Arduino arenduskeskkonna vaja minevate elementide kirjeldus

Seejärel ongi arendusplaat programmeeritud, järgnevas peatükis kirjeldatakse täpsemalt, millist koodi arendusplaadile laadida.

### **Madala sinihamba energia laiendusplaadi ühendamine arendusplaadiga**

Madala sinihamba energia laiendusplaadi ja Iteduino UNO arendusplaadi ühendamiseks kasutati internetis avaldatud temperatuuri mõõtmise näidet, mis on saadaval lehel <http://agostini.tech/2017/02/20/creating-a-temperature-sensor-for-ios-using-ble-and-arduino/>. Näites on joonis, kuidas ühendada viikuseid Arduino Uno arendusplaadi ja

sinihamba laiendusplaadi vahel (vt. Joonis 8). Plaatide ühendamiseks on vaja kaheksat juhet ning need peab ühendama samamoodi nagu on joonisel (vt. Joonis 8) näidatud.



Joonis 8. Arendusplaadi ning sinihamba laiendusplaadi ühendamine viikudega [11]

Näites on kirjeldatud, kuidas Arduino UNO arendusplaati programmeerida nii, et sinihamba signaal oleks kättesaadav seadmetes. Näites toodud koodis eemaldatakse temperatuuri saatmisega seotud funktsioonid *averageTemperature*, *averageValue* ning *temperature*. Seejärel muudetakse muutuja *temperatureString* nimi *sobivamaks näiteks scaleString*. Ning muutuja väärtustus asendatakse tühja sõnega ja mujal muutuja välja kutsumised asendatakse uue nimega. Sinihamba signaali nimi, mida kuvatakse seadmes kui signaali otsitakse, muudetakse ära. *BLETemp* asendatakse saab *BLEScale* nimega. Lõplik muudatustega kood on kuvatud joonisel (vt. Joonis 9).

```
#include "Adafruit_BLE_UART.h"

#define ADAFRUITBLE_REQ 10
#define ADAFRUITBLE_RDY 2
#define ADAFRUITBLE_RST 9

Adafruit_BLE_UART BTLEserial = Adafruit_BLE_UART(ADAFRUITBLE_REQ, ADAFRUITBLE_RDY,
ADAFRUITBLE_RST);

const int numReadings = 50;
const int initialValue = 144;
```

```

int readings[numReadings];
int readIndex = 0;
int total = initialValue * numReadings;

const int sensorPin = A0;

void setup() {
  Serial.begin(9600);

  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = initialValue;
  }

  BTLEserial.setDeviceName("BLEScale");
  BTLEserial.begin();
}

aci_evt_opcode_t laststatus = ACI_EVT_DISCONNECTED;

void loop() {
  BTLEserial.pollACI();

  aci_evt_opcode_t status = BTLEserial.getState();
  if (status != laststatus) {
    laststatus = status;
  }

  if (status == ACI_EVT_CONNECTED) {
    String scaleString = "";

    uint8_t sendbuffer[20];
    scaleString.getBytes(sendbuffer, 20);
    char sendbuffersize = min(20, scaleString.length());

    Serial.print(F("\n* Sending -> \")); Serial.print((char *)sendbuffer);
    Serial.println("\");

    BTLEserial.write(sendbuffer, sendbuffersize);
  }
}

```

Joonis 9. Madala sinihamba energia laiendusplaadi ühendamine Iteaduno UNOga kood

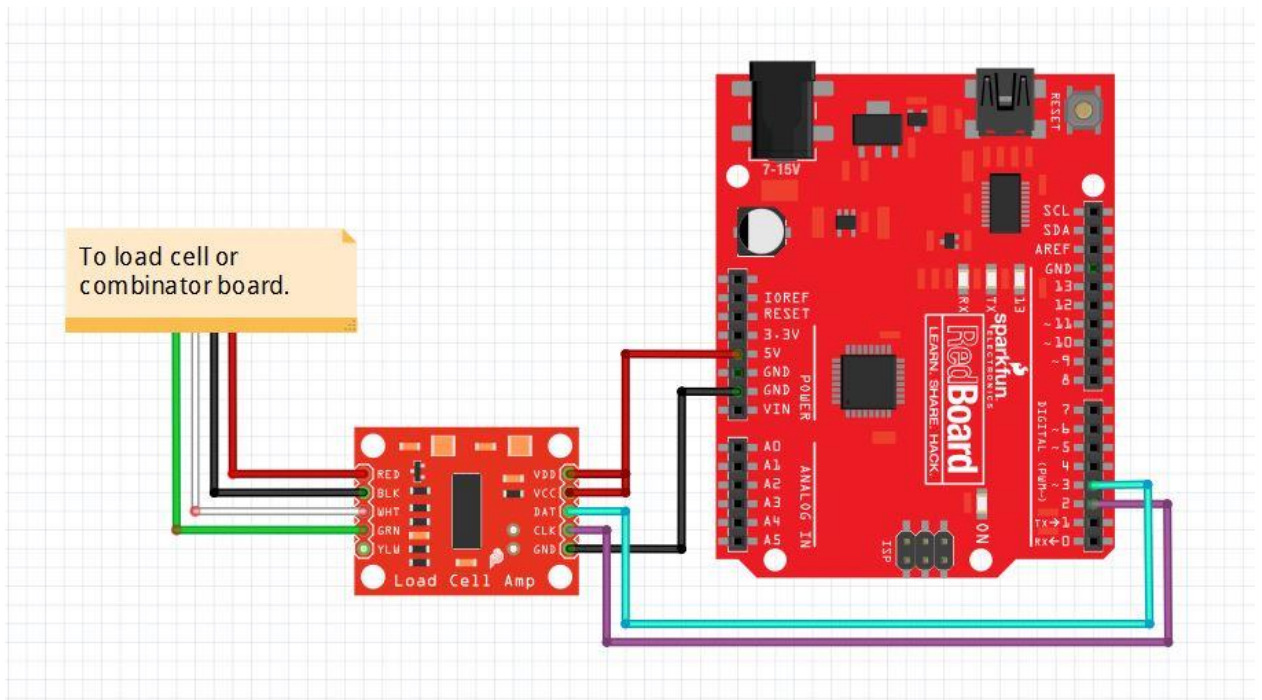
Sinihamba signaali testimiseks on võimalik kasutada mobiilirakendust Bluefruit, mida on võimalik laadida alla App Storest. Tihtipeale seadme sisene sinihamba signaali otsija ei kuva Adafruit sinihamba laiendusplaadi signaali nime, kuid spetsiaalne rakendus leiab selle signaali ilma probleemideta üles.

### **Kaalu näidu saatmine Arduino UNO arendusplaati**

Kaalu ühendamiseks arendusplaadiga kasutati näidet, kus on ära kirjeldatud kuidas viikused ühendada ning on antud Arduino kood. Näite leiab interneti leheküljelt <https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide>.

Viikude ühendamisel analoog-digitaalmuunduriga pidi erinevat värvi juhtmed teistmoodi panema kui näites oli kirjeldatud, sest käesolevas töös kasutatud kaalumõõtmise elemendi valmistamisel olid juhtmete värvid segamini aetud. Teise tootja poolt katsetamiseks ostetud

sensori ühendamisel analoog-digitaalmuunduriga sai täpselt jälgida näites toodud joonist (vt. Joonis 10). Seda sensorit ei kasutata käesolevas töös kuna tema kandevõime on ainult üks kilogramm.



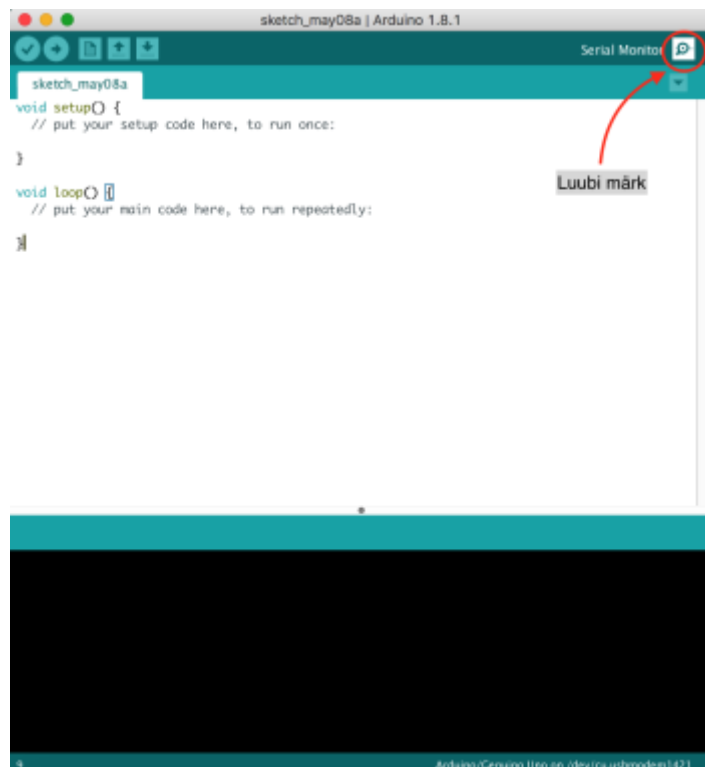
Joonis 10. Kaalu mõõtmise elemendi, HX711 analoog-digitaalmuunduri ja Iteduino UNO omavahel ühendamise juhtmetega [12]

Kaalunäidu nägemiseks tuleb Arduino arenduskeskkonnas vajutada üleval paremal luubi märgile (ingl.k. serial monitor) (vt. Joonis 11) avanevas aknas kuvatakse programmi väljund/väljundid. Esialgsel ühendamisel on kaalunäit vale, seega on vajalik kaalu kalibreerida ehk kaalunäitu reguleerida. Kaalu kalibreerimiseks kasutatakse koodi, mis on saadaval interneti leheküljelt <https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide>. Näites on kasutatud kaalunäidu mõõteühikuks naela ehk massiühikut inglise mõõdusüsteemis, seega kalibreerimise tegur kilogrammide jaoks erineb oluliselt näite omale. Näites toodud koodis muudetakse ära ühiku sõne *lbs* kilogrammi lühendi sõne *kg* vastu (vt. Joonis 12). Ning samuti modifitseeritakse kaalu täpsust kolme kohani pärast koma, ehk real `Serial.print(scale.get_units(),1)` kirjutatakse parameetri 1 asemel 3 (vt. Joonis 12). Seejärel laetakse muudetud kood arendusplaadile ning plaat jäetakse arvutiga ühendatuks. Nüüd võib hakata kaalu kalibreerima. Kaalu tulemuste nägemiseks vajutatakse luubi märgile (vt. Joonis 11)

Kalibreerimise sammud:

1. kaaluda objekt mõnel olemas oleval toidukaalul;
2. kirjutada objekti kaal paberile või mujale üles;
3. asetada objekt kaalu mõõtmise anduri korpusele;
4. kaalu tulemuste aknas sisestada + või - märki nii kaua kuni kaalunäidu tulemus on õige (märkide sisestamine modifitseerib kalibreerimise tegurit, mis on kaalu tulemuste aknas näha);

Kalibreerimise samme korratakse mitu korda erinevate objektidega, et kalibreerimise tegur oleks võimalikult täpne. Pärast kaalunäidu reguleerimist jäetakse meelde uus kalibreerimistegur.



Joonis 11. Luubi märgi (ingl.k. serial monitor) asukoht Arduino arenduskeskkonnas

```
/*  
Example using the SparkFun HX711 breakout board with a scale  
By: Nathan Seidle  
SparkFun Electronics  
Date: November 19th, 2014  
License: This code is public domain but you buy me a beer if you use this and we  
meet someday (Beerware license).  
*/  
#include "HX711.h"  
  
#define DOUT 3  
#define CLK 2
```

```

HX711 scale(DOUT, CLK);

float calibration_factor = -7050; //-7050 worked for my 440lb max scale setup

void setup() {
  Serial.begin(9600);
  Serial.println("HX711 calibration sketch");
  Serial.println("Remove all weight from scale");
  Serial.println("After readings begin, place known weight on scale");
  Serial.println("Press + or a to increase calibration factor");
  Serial.println("Press - or z to decrease calibration factor");

  scale.set_scale();
  scale.tare(); //Reset the scale to 0

  long zero_factor = scale.read_average(); //Get a baseline reading
  Serial.print("Zero factor: "); //This can be used to remove the need to tare the
scale. Useful in permanent scale projects.
  Serial.println(zero_factor);
}

void loop() {

  scale.set_scale(calibration_factor); //Adjust to this calibration factor

  Serial.print("Reading: ");
  Serial.print(scale.get_units(), 1);
  Serial.print(" kg"); //Change this to kg and re-adjust the calibration factor if
you follow SI units like a sane person
  Serial.print(" calibration_factor: ");
  Serial.print(calibration_factor);
  Serial.println();

  if(Serial.available())
  {
    char temp = Serial.read();
    if(temp == '+' || temp == 'a')
      calibration_factor += 10;
    else if(temp == '-' || temp == 'z')
      calibration_factor -= 10;
  }
}

```

Joonis 12. Kaalu kalibreerimise Arduino kood

Nüüd on vaja eemaldada koodist kalibreerimisega seotud abikood, et kood oleks loetav ning puhas. Lisaks tuleb muuta kalibreerimise teguri *calibration\_factor* väärtust vastavalt eelnevalt saadud tulemusele. Lõplik kaalunäidu kätte saamise kood on nähtav joonisel (vt. Joonis 13).

```

/*
Example using the SparkFun HX711 breakout board with a scale
By: Nathan Seidle
SparkFun Electronics
Date: November 19th, 2014
License: This code is public domain but you buy me a beer if you use this and we
meet someday (Beerware license).
*/
#include "HX711.h"

#define calibration_factor -500360 //This value is obtained using the
SparkFun_HX711_Calibration sketch

```

```

#define DOUT 3
#define CLK 2

HX711 scale(DOUT, CLK);

void setup() {
  Serial.begin(9600);
  Serial.println("HX711 scale demo");

  scale.set_scale(calibration_factor); //This value is obtained by using the
SparkFun_HX711_Calibration sketch
  scale.tare(); //Assuming there is no weight on the scale at start up, reset the
scale to 0

  Serial.println("Readings:");
}

void loop() {
  Serial.print("Reading: ");
  Serial.print(scale.get_units(), 3); //scale.get_units() returns a float
  Serial.print(" kg"); //You can change this to kg but you'll need to refactor the
calibration_factor
  Serial.println();
}

```

Joonis 13. Kaalunäidu kättesaamise Arduino kood

Nüüd on teada, kuidas arendusplaadil kätte saada kaalunäitu ning kuidas teha sinihamba signaali iOS operatsioonisüsteemiga seadme jaoks nähtavaks.

### Arendusplaadile laetav lõplik kood

Kaalunäidu saatmiseks seadmele on vaja eelneva kahe alampeatüki Arduino koodid ühendada ning saadud tulemus laadida arendusplaadile. Madala sinihamba ühendamisel arendusplaadiga koodis tuleb nüüd *scaleString* tühi väärtustus asendada kaalunäidu saamise operatsioonidega. Ühendatud kood on nähtav joonisel (vt. Joonis 14. Kaalunäidu saatmise sinihamba signaaliga Arduino kood).

```

#include <SPI.h>
#include "Adafruit_BLE_UART.h"
#include "HX711.h"

#define ADAFRUITBLE_REQ 10
#define ADAFRUITBLE_RDY 2
#define ADAFRUITBLE_RST 9

#define calibration_factor -500360 //This value is obtained using the
SparkFun_HX711_Calibration sketch
#define DOUT 4
#define CLK 3

HX711 scale(DOUT, CLK);

Adafruit_BLE_UART BTLEserial = Adafruit_BLE_UART(ADAFRUITBLE_REQ, ADAFRUITBLE_RDY,
ADAFRUITBLE_RST);

const int numReadings = 50;
const int initialValue = 144;

```

```

int readings[numReadings];
int readIndex = 0;
int total = initialValue * numReadings;

const int sensorPin = A0;

void setup() {
  Serial.begin(9600);

  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = initialValue;
  }
  scale.set_scale(calibration_factor); //This value is obtained by using the
SparkFun_HX711_Calibration sketch
  scale.tare(); //Assuming there is no weight on the scale at start up, reset the
scale to 0

  BTLEserial.setDeviceName("BLEScale");
  BTLEserial.begin();
}

aci_evt_opcode_t laststatus = ACI_EVT_DISCONNECTED;

void loop() {

  BTLEserial.pollACI();

  aci_evt_opcode_t status = BTLEserial.getState();
  if (status != laststatus) {
    laststatus = status;
  }
  Serial.print(status == ACI_EVT_CONNECTED);

  if (status == ACI_EVT_CONNECTED) {

    String scaleString = getScaleUnits();
    Serial.print("Connected");
    Serial.print(scaleString);

    uint8_t sendbuffer[20];
    scaleString.getBytes(sendbuffer, 20);
    char sendbuffersize = min(20, scaleString.length());

    Serial.print(F("\n* Sending -> \"")); Serial.print((char *)sendbuffer);
    Serial.println("\"");

    BTLEserial.write(sendbuffer, sendbuffersize);
  }
}

String getScaleUnits() {
  return String(scale.get_units(),3);
}

```

Joonis 14. Kaalunäidu saatmise sinihamba signaaliga Arduino kood

Pärast koodi laadimist arendusplaadile on toidukaal valmis. Järgmisena saab hakata arendama kaaluga ühilduvat rakendust.

### **3. Targa toidukaalu mobiilirakendus**

Mobiilirakendusi, mis ühilduvad sinihamba toidukaaluga on juba mitmeid, kuid kõik neist ei ole eestikeelsed ning ei sisalda enamike Eesti poodides leiduvaid tooteid. Seega hetkel turul olevad rakendused ei sobi Eestis elava inimese toidulaua jälgimiseks.

#### **3.1 Visioon**

Targa toidukaalu rakendus aitab Eestis elavatel inimesel lihtsamini jälgida enda toitumist. Rakenduse saab ühildada sinihamba ehk traadita ühenduse toidukaaluga, mis edastab mobiilirakendusele kaalunäidu. Rakenduse peamiseks eesmärgiks on kiirelt saada informatsiooni toidukaalule asetatud toiduaine kohta, nagu näiteks kilokalorite, valkude, rasvade ja süsivesikute sisaldus. Enamik Eestis müüdavad toiduained peaksid olema mobiilirakenduses leitavad. Toitumise mugavamaks jälgimiseks on võimalik lisada toitude informatsioon päevasesse toitumiskavasse. Kasutajal on võimalik näha kalorite, valkude, rasvade ja süsivesikute hulka mingil päeval.

#### **3.2 Kasutatavad tehnoloogiad**

Targa toidukaalu rakendus luuakse operatsioonisüsteemile iOS 10 ja uuematele versioonidele. Apple toodete eestlaste kasutajaskonna kujund on terviseteadlikum ning autor kasutab ise Apple tooteid ning seega otsustas autor teha rakenduse iOS operatsioonisüsteemile. Mobiilirakenduse arendamiseks kasutatakse populaarset programmeerimiskeelt Swift, mis on mõeldud iOS operatsioonisüsteemiga seadmete rakenduste arendamiseks. Autor ei valinud Objective-C programmeerimiskeelt kuna oli varasemalt kokku puutunud Swift keelega. Arendamisel kasutatakse integreeritud programmeerimiskeskonda xCode, mis on saadaval ainult macOS operatsiooni süsteemiga arvutitele. Keskkond on arendajate seas väga populaarne ning paljud õpetused on tehtud kasutades xCode keskkonda ning seega on seda kõige mõistlikum kasutada.

Mobiilirakenduses kasutatakse JTAppleCalendar rakenduse raamistiku (ingl.k. application framework), sest Swift programmeerimiskeelel ei ole tehtud eraldi kalendri funktsionaalsust ning mõttekas on kasutada mõnda olemas olevat raamistiku [13].

#### **3.3 Olemasolevate toodete kirjeldus**

Järgnevalt võrreldakse tabelis Tabel 3 kahte turul müüdavat inglise keelse kasutajaliidesega targa toidukaalu rakendust Situ [14] ja Smart Diet Scale [15].

Tabel 3. Situ ja Smart Diet Scale toodete võrdlus

Tootemark	Situ	Smart Diet Scale
Toidukaal	Sinihamba toidukaal	Sinihamba kaal, võimalusega kaaluda nelja erinevat toitu korraga
Toidukaalu maksumus	89.99 eurot	118,72 eurot
Mobiilirakenduse maksumus	Tasuta	Tasuta
Toitude andmebaas	Suurbritannia ning Ameerika Ühendriikides müüdavad toidud	Ameerika Ühendriikides müüdavad toidud
Operatsioonisüsteemi ühilduvus	iOS	Android ja iOS

Kumbki toodetest ei ole Eestis elavale inimesele kasutamiseks sobilikud kuna rakendustes puuduvad paljud Eestis müüdavad toidud.

### 3.4 Nõuete Analüüs

Mobiilirakenduse loomiseks on esmalt vaja teostada nõuete analüüs, see annab hea ülevaate mida rakenduses peaks kajastama ning kuidas seda teha. Funktsionaalsed nõuded määravad ära süsteemi oodatava käitumise ning mitte-funktsionaalsed nõuded kirjeldavad kvaliteeti, mida programm järgima peaks [16].

#### Funktsionaalsed nõuded

Funktsionaalsed nõuded on kategoriseeritud teemade kaupa, et oleks lihtsam järele pidada. Ning iga nõude juurde on lisatud kirjeldus.

### ***Ekraanivaated***

Rakenduses on kolm ekraanivaadet: kaalunäidu, toitude ning päeviku vaated. Kaalunäidu ehk esimesest vaatest saab liikuda teistesse vaadetesse. Toite on võimalik näha vajutades luubimärgile ning päeviku vaatesse saab vajutades menüüs *Päevikule*. Toitude ning päeviku vaadetes on olemas tagasiminemise nupp, mida vajutades avaneb kaalunäidu vaade.

### ***Kaalunäidu ekraanivaade***

Rakendus näitab ühendatud sinihamba toidukaalu kaalunäitu. Esialgu on kaalunäit punast värvi, kuid pärast kaalu ühendamist muutub see mustaks. Rakenduses kasutatakse mõõtühikute SI süsteemi ehk rahvusvahelist mõõtühikute süsteemi ning kuvatakse kaalunäit kilogrammides.

### ***Toitude ekraanivaade***

Toidud on kuvatud tabeline, kus reale vajutades avaneb täpsem informatsioon. Toitude sorteerimiseks on võimalik kasutada otsingumootorit, kuhu tuleb sisestada märksõnu ning seejärel tabelis uuendatakse tulemused vastavalt sisestatud sõnale. Kasutajal on võimalik vajutada toidu real olevale plussmärgile, mis lisab toidu ning informatsiooni päevikusse. Pärast plussmärgi vajutamist avaneb taas kaalunäidu ekraanivaade.

### ***Päeviku ekraanivaade***

Päeviku ekraanivaates kuvatakse kalender, mis on avatud hetkese kuupäevaga. Kalendris kuupäevadele vajutades avaneb kalendri all tabel, kus on söödud toidud ning nende informatsioon.

### **Mittefunktsionaalsed nõuded**

Mittefunktsionaalsed nõuded on kuvatud listina ning iga nõude juurde on lisatud kirjeldus.

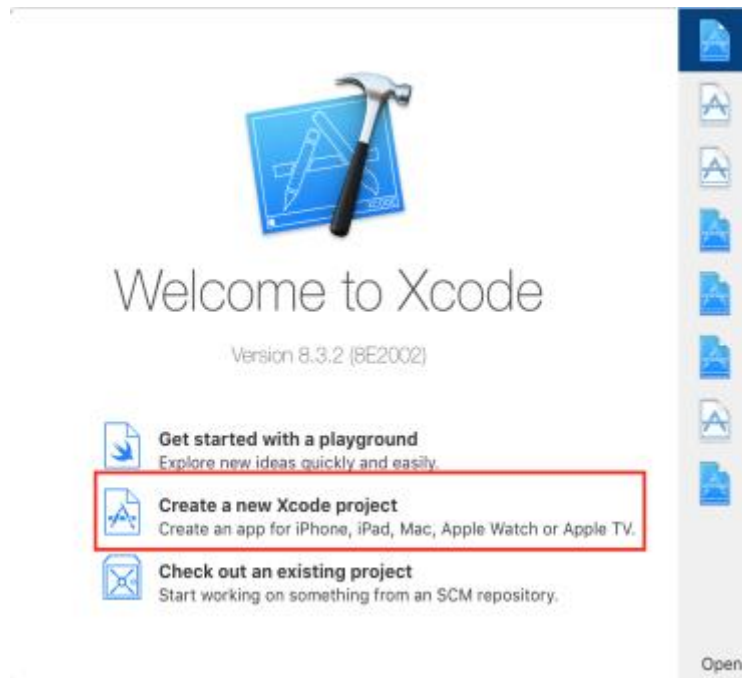
1. Rakenduses on vähemalt 4000 erinevat toitu, mis tagab, et rakenduses on enamus Eestis müüdavatest toitudest.
2. Rakendus ühendub automaatselt sinihamba toidukaaluga kui sinihammas on seadmes sisse lülitatud. Nõue tagab selle, et ei tekiks probleeme kaalu käsitsi ühendamisel rakendusega.
3. Rakendus on eestikeelne. Rakendus on mõeldud Eestis elavale inimesele kasutamiseks ning suurem osa Eestis elavatest inimestest saab aru eesti keelest.
4. Rakenduse andmed talletakse kasutaja seadmes. Rakendusel puudub sisse logimise süsteem ning seega ei ole mõttekas toitumise andmeid hoida andmebaasis.

5. Kaalunäit kuvatakse ümardatuna kolmekohani pärast koma.
6. Rakenduse kõik funktsionaalsused töötavad ilma interneti ühenduseta.

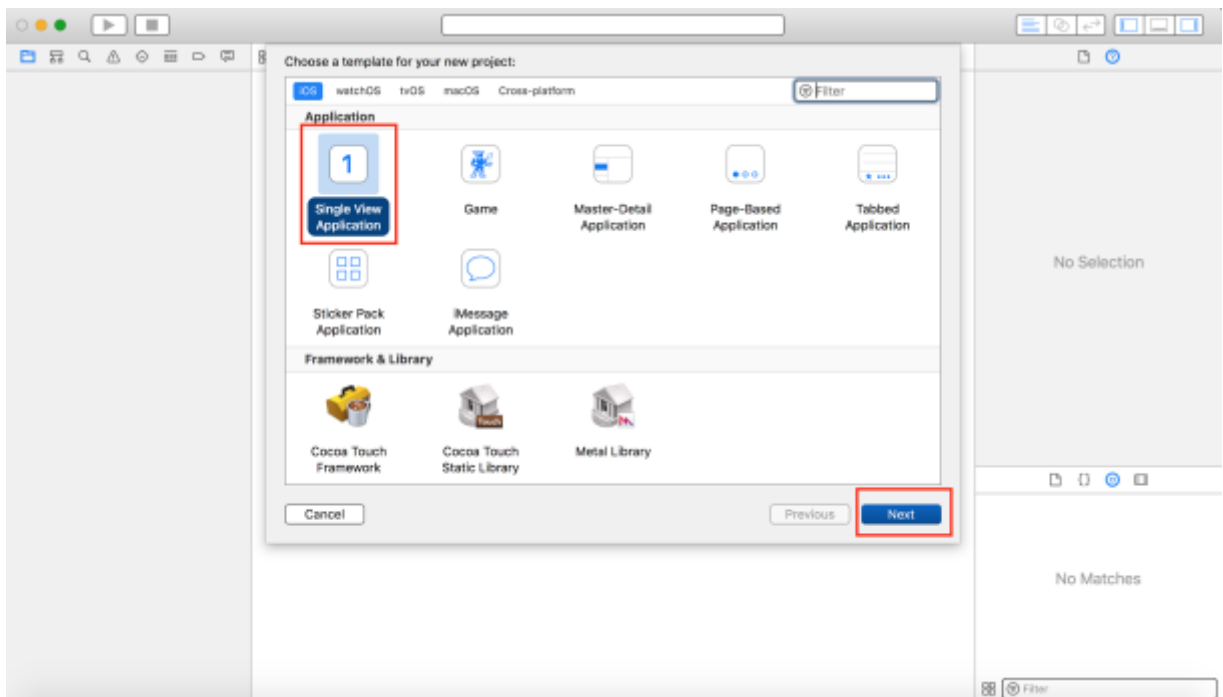
Funktsionaalsete nõuete täitmiseks on tihtipeale vaja tehnoloogiaid, mis on kirjeldatud järgmises peatükis.

### 3.5 Sinihamba kaalunäidu kuvamine rakenduses

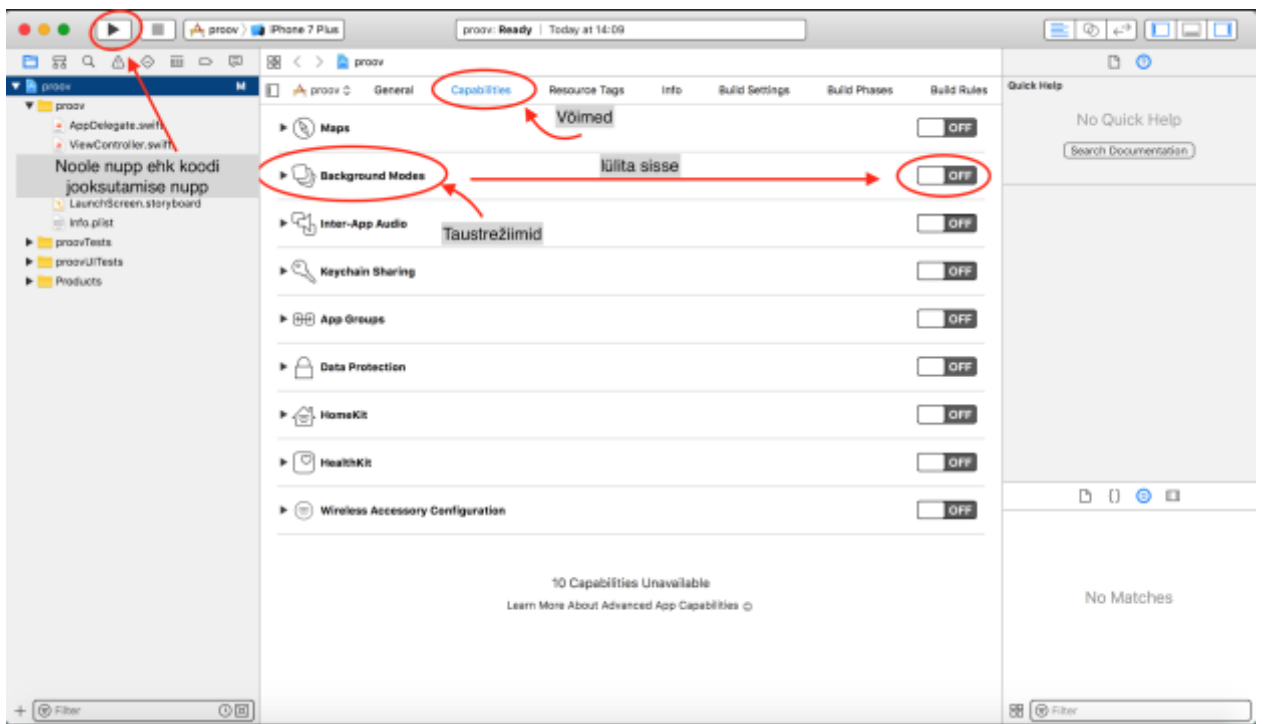
Mobiilirakenduse arendamiseks kasutatakse juba eelnevalt mainitud temperatuuri saatmise näidet, mis on leitav interneti leheküljel <http://agostini.tech/2017/02/20/creating-a-temperature-sensor-for-ios-using-ble-and-arduino/>. Näites on programmeerimiskeeles Swift kood, mis leiab üles sinihamba signaali ning saab kätte temperatuuri näidu. Mobiilirakenduse arendamiseks kasutatakse integreeritud programmeerimiskeskonda xCode, mis on saadaval ainult macOS operatsioonisüsteemiga arvutitele. Keskkonna saab alla laadida ametlikult Apple kodulehelt <https://developer.apple.com/download/>. Programmi avades tuleb valida *Create new xCode project* (vt. Joonis 15) ning siis valida *Single View Application* ja seejärel *next* (vt. Joonis 16). Avanevas aknas tuleb *product name* välja lisada projekti nimi ning programmeerimise keele valikust *language* valida Swift. Avanevas aknas valida asukoht kuhu projekti kaust salvestatakse ning valida *create*. Nüüd peab seadistama projektis sinihamba signaali lubamise. Valides menüüst *capabilities* ning seejärel peab *background modes* sisse lülitama (vt. Joonis 17). Järgmisena peab linnukese panema *uses Bluetooth LE accessories* ning sellega ongi sinihamba signaal lubatud projektis. Järgnevalt on vaja luua kaks faili .swift laiendiga nimedega *ViewController* ning *BLEManager* kausta, mille nimi on eelnevat kirjutatud projekti nimi. Näites olevad koodi lõigud on vaja kopeerida vastavate failide sisse ning siis on võimalik koodi jooksutada vajutades noole nupule (vt. Joonis 17).



Joonis 15. xCode keskkonnas uue projekti tegemine



Joonis 16. Rakenduse tüübi valimine xCodes

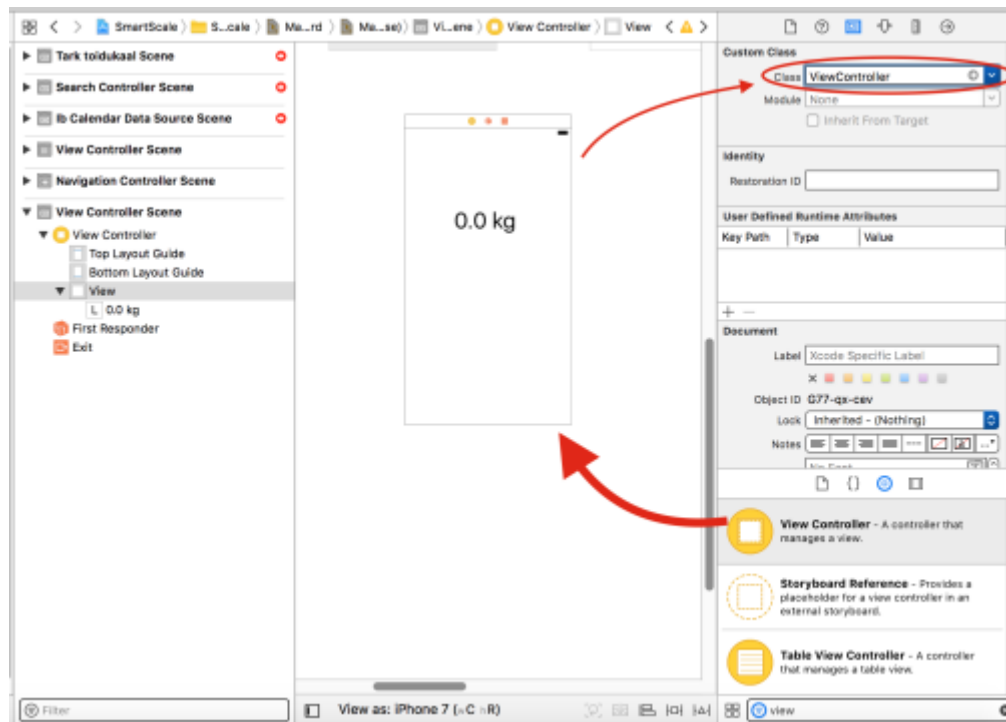


Joonis 17. xCode projekti taustrežiimide sisse lülitamine

Koodi jooksutades ei ilmu simulaatori ekraanile peale valge tausta midagi, sest kaalunäit ei ole seadistatud kuvamiseks.

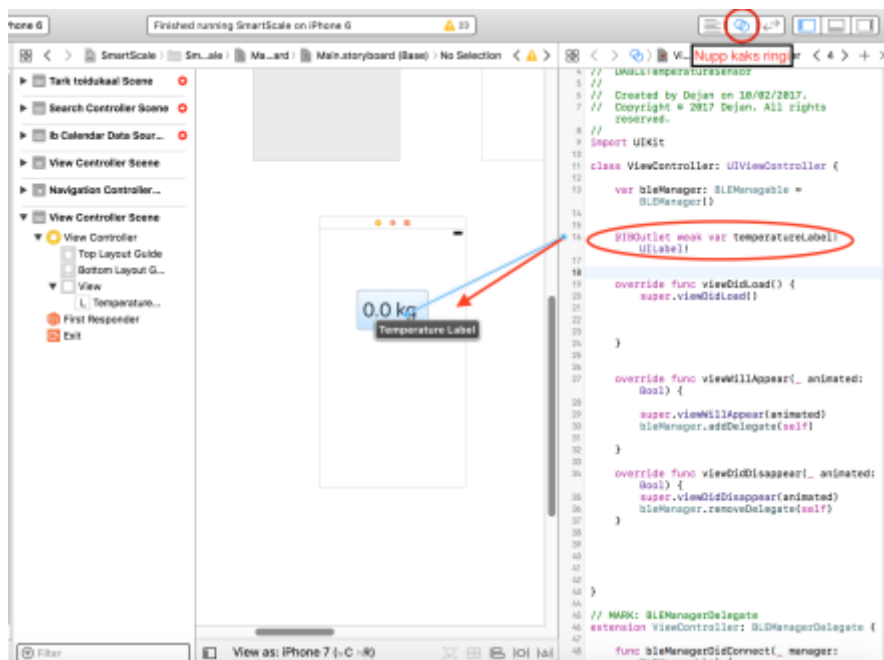
### ***Kaalunäidu kuvamine ekraanivaatel***

Eelmise alampeatüki koodi jooksutades ei ilmunud simulaatorile kaalunäitu. Kaalunäidu näitamiseks on programmeerimiskeskkonnas võimalik kasutada süžeetahvleid (ingl.k. storyboard) kasutajaliidese arendamiseks. Süžeetahvli failis *Main.storyboard* on vaja luua stseenid. Stseenid ja selle elemendid peab ühendama koodiga. *ViewController.swift* fail on vaja ühendada vaate stseeniga. Süžeetahvli failis on vaja lohistada element kuva kontrolleri (ingl.k. View Controller) tahvlile ning muuta kontrolleri elemendi klassi nimi (vt. Joonis 18). Elemendi klassi nimi on failis *ViewController*, kuna näite koodijupis oli klass sellise nimega defineeritud. Järgnevalt on vaja lohistada element silt (ingl.k. label) kuva kontrolleri sisse ning seejärel muuta sildi tekst *0.0 kg* sõneks.



Joonis 18. Süžeetahvli faili sisu koostamine xCodes

Nüüd on vaja süžeetahvli silt ühendada koodiga. Programmeerimiskeskkonnas on vaja avada vaade, kus on võimalik näha süžeetahvli faili ja koodi faili korraga, selleks on vaja vajutada nupule kaks ringi, mis on kuvatud ka joonisel (vt. Joonis 19). Seejärel on vaja teha kindlaks, et õige koodi fail avanes, vajadusel saab faili muuta ülevalt kahe ringi nupu alt. Järgmisena peab koodi faili temperatuuri sildi juures olevast ringist tõmbama hiirega joone süžeetahvli faili sildile nii nagu on näidatud joonisel (vt. Joonis 19). Nüüd on kõik vajalikud ühendused koodi ja süžeetahvli failide vahel tehtud. Arusaadava koodi huvides peaks temperatuuri sildi muutuja nime ära vahetama.



Joonis 19. Süžeetahvli stseeni elemendi ühendamine koodiga xCodes

Koodi jooksutades ilmub simulaatori ekraanile kaalu silt *0.0 kg*. Selleks, et sildi asemele tekiks näit, mis saadakse ehitatud kaalust, on vaja rakendust käivitada seadmel.

### ***Kaalunäidu kuvamise testimine iOS seadmel***

Integreeritud programmeerimiskeskond xCode lubab jooksutada koodi simulaatoritel, kuid sinihamba signaali katsetamiseks on vajalik jooksutada koodi seadmes. Arvutis simulaatoril koodi jooksutades ei leita sinihamba signaali üles kuna arvutil on ainult üks madala energia sinihamba signaali otsija ning arvuti ei luba seda hõivata simulaatori jaoks. Arvutile on võimalik osta lisaks sinihamba madala energia USB vastuvõtja ning sel juhul saab rakendust testida simulaatoril. Rakenduse testimiseks seadmel on vaja kõigepealt xCode keskkonnas projekt seadistada, selleks tuleb vajutada failihalduris projekti nimele ning seejärel avada menüüst *General*. Avanenud seadetes on vaja lisada unikaalne kimbu identifikaator (ingl. k. bundle identifier), mis võib olla suvaline, kuid peab olema unikaalne. Ning seejärel on tiimi (ingl. k. team) alt vaja lisada Apple ID kasutaja, mida kasutatakse seadmes, kus tahetakse rakendust testida. Nüüd tuleb iOS seade ühendada juhtmega arvuti külge ning jooksutades valida sihtmärgiks antud seade. Seadmes avatakse automaatselt rakendus ning lülitades sisse sinihamba on võimalik näha kaalunäidu kuvamist.

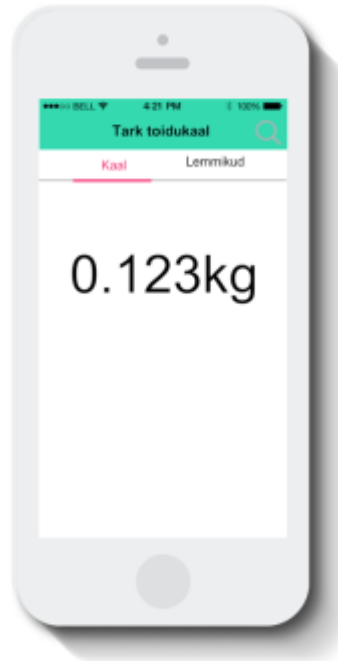
Pärast kaalunäidu kuvamise testimist on võimalik mobiilirakendust edasi arendada.

### 3.6 Mobiilirakenduse kasutajaliides

Rakenduse edasiseks arendamiseks on vaja funktsionaalsused kujutada joonistel, et oleks parem aimdus, kuidas rakendust programmeerida. Kasutajaliidesel on kolm erinevat vaadet, mis on kirjeldatud teksti ja joonisena.

#### Põhivaade

Põhivaadet illustreerib joonis (vt. Joonis 20).

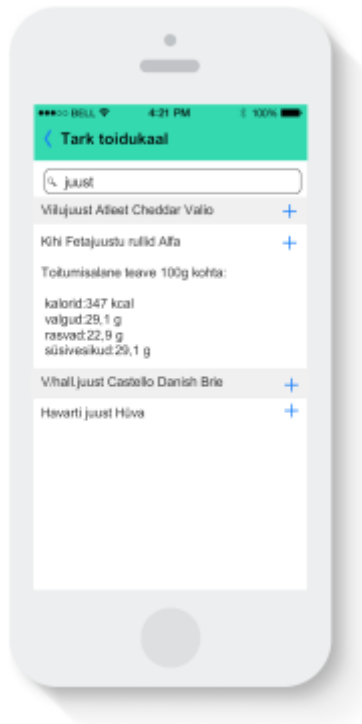


Joonis 20. Põhivaade

Kaalunäit kuvatakse vastavalt sinihamba kaalunäidule, kui kaal ei ole ühendatud on näidu tekst punane. Vajutades üleval paremal luubi märgile avaneb otsinguvaade (vt. Joonis 21) ning salvestatakse viimane kaalunäit.

#### Otsinguvaade

Otsinguvaadet illustreerib joonis (vt. Joonis 21). Seal on otsingusse kirjutatud “juust” ning otsingu all on kuvatud tulemused. Vajutades reale avaneb toidu kohta toitumisalane informatsioon, joonisel (vt. Joonis 21) on näha Kihi Fetajuustu rulli Alfa toitumisalane teave.



Joonis 21. Otsinguvaade

Vajutades real pluss märgile saab lisada vastava toidu päevikusse tänasele kuupäevale koos toitumisalase teabega vastavalt kaalunäidule (vt. Joonis 22).

### **Päeviku vaade**

Päevikuvaates kuvatakse kalender, mis on avatud automaatselt tänase kuupäevaga (vt. Joonis 22). Vajutades mõne kuupäeva peale on kalendri all kuvatud tabel lisatud toitude ja toitumisalase informatsiooniga vastavalt kuupäevale.



Joonis 22. Päeviku vaade

Kasutajaliides on koostatud jälgides lihtsuse põhimõtet. Pärast rakenduse avamist kuvatakse kasutajale koheselt kaalunäit, mis on üks tähtsamaid osasid rakenduse juures. Muud liigset informatsiooni ei ole kaalunäidu juurde pandud, sest inimese silm on harjunud tavalisel toidukaalul nägema ainult kaalunäitu. Päevikus kasutatakse kalendrit, kuna inimesed on harjunud sellise formaadiga ning kalendris on võimalik väga kiiresti ja hõlpsasti kuupäevade vahel liikuda. Toite kuvatakse tabelina kuna sel viisil on kerge lugeda loetelu, nende kuvamiseks rakenduses on esmalt vaja toitude nimekirja.

### 3.7 Toidud rakenduses

Rakenduse üks põhilisemaid eesmärke on pakkuda kasutajatele relevantseid tooteid, et rakendust oleks mõttekas Eestis elaval inimesel kasutada. Eestis paljudel toitu müüvatel kaubanduskettidel on virtuaalsed poed. Autor kirjutas mitmetele kaubandusketi esindajatele ning uuris, et ka oleks võimalik ligi pääseda virtuaalse poe toodete andmetele. Positiivne vastus tuli Coop Eesti jaekaubanduse grupilt. Mobiilirakenduses kasutatakse Coopilt saadud toite koos toitumisalase informatsiooniga. Toodete listis on 4310 erinevat toitu ning need katavad enamuse Eestis müüdavate toodete hulga. Kõige uuemaid tooted ei ole kättesaadavad rakenduses kuna toitude list koostati enne 2016 aasta detsembrit.

### 3.8 Kalendri kuvamine

Rakenduses on võimalik lisada toitusid päevikusse, mis kuvatakse kalendrina. Kalendri kuvamisel kasutatakse JTAppleCalendar rakenduse raamistikku. Raamistikku kasutades on võimalik kujundada kalendri välimust vastavalt soovidele ning seetõttu on raamistiku sobilik rakenduse arendamisele.

Raamistik lisatakse kasutades Carthage sõltuvuse haldurit. Carthage saab alla laadida GitHub koodirepositooriumist <https://github.com/Carthage/Carthage/releases> vajutades *Carthage.pkg* lingile. Kui fail on alla laetud siis peab selle avama ning järgima juhiseid, seejärel on Carthage alla laetud. Nüüd saab lisada JTAppleCalendar raamistiku rakendusele. Kõige lihtsam on selleks järgida video näidet <https://www.youtube.com/watch?v=SSFxzNXzn2k>, kus näidatakse kuidas lisada raamistikku projektile.

Kalendri loomisel on vajalik kasutada süžeetahvleid ning süžeetahvli elemente on vaja ettevaatlikult ühendada koodiga, et kood kompileeriks. Kalendri valmistamiseks on olemas video näide <https://www.youtube.com/watch?v=zOphH-h-qCs>. Ning kalendri disainimiseks kasutatakse video näidet [https://www.youtube.com/watch?v=Qd\\_Gc67xzlW&t=905s](https://www.youtube.com/watch?v=Qd_Gc67xzlW&t=905s), kus näidatakse lihtsa ent hea välimusega kalendri disainimist.

Eelnevalt mainitud kalendri valmistamise video näidetes on kasutatud kalendrit, mis algab pühapäevast ning lõppeb laupäevaga. Eestis kasutatakse laialdaselt aga kalendreid, mis algavad esmaspäevaga ning lõppevad pühapäevaga. Kuupäevade korrektseks kuvamiseks esmaspäevaga algavast kalendrit oleks vaja esmalt muuta süžeetahvlis siltide järjekorda nii, et esimene ehk pühapäeva silt läheb viimaseks. Seejärel on vaja koodis täpsustada, et kalender algab esmaspäevast. Seega on vaja lisada funktsiooni *configureCalendar* muutuja *parameters* väärtusele *ConfigurationParameters(startDate:startDate,endDate:endDate)* sulgudesse komaga eraldatult juurde parameeter *firstDayOfWeek:DaysOfWeek.monday*. Lisatud parameeter kirjeldabki ära, mis päevast algab kalender ning inglise keelne sõna *monday* tähendab esmaspäeva.

Praegu avaneb kalender jaanuarist. Kalendri avamiseks käesoleva kuupäevaga peab tegema koodis mõningaid muudatusi. Selleks peab lisama funktsiooni *viewDidLoad* kaks rida. Esimesena peaks lisama rea *calendar.scrollToDate(Date())*, mis avab kalendri õige kuuga, ning parameeter *Date()* tähistab hetke kuupäeva. Ning teise reana lisatakse

`calendar.selectDates([Date()])`, mis valib praeguse kuupäeva. `SelectDates` funktsioon nõuab parameetriks listi, kuna funktsioon lubab tähistada mitu kuupäevaga korraga.

Projekti paremaks ülevaateks on lisatud tööle failide arhitektuur, mis kirjeldab rakenduse ülesehitust.

### 3.9 Lahenduse failide arhitektuur

Projekti põhilised failid asuvad kasutas `SmartScale`. Kuna ekraanivaateid on rakenduses kolm, siis on kaustas kolm vaate kontrolleri faili `ViewController` ehk põhivaade, `SearchController` ehk otsingu vaade ning `CalendarViewController` ehk päeviku vaade. Lisaks on kaustas süžeetahvli failid `Main.storyboard`, kus kirjeldatakse visuaalselt projekti kasutajaliides ning `LaunchScreen.storyboard`, kus kirjeldatakse rakenduse laadimisel kuvatav ekraanivaade. Kaustas leidub ka `BLEManager.swift` fail, kus on sinihamba kaaluga ühendamiseks vajalikud funktsioonid. Lisaks on projekti kaustas abiklassid, mis aitavad vaateid kuvada. Programmeerimiskeskonnas xCode projekti luues, tehakse veel testimise ja toote kaustad, mida käesolevas töös ei modifitseeritud. Lisaks on projektis toitute teksti ning kalendri raamistiku failid.

Rakenduse arendamise protsessiks vajalikud komponendid on nüüd kirjeldatud. Järgnevalt kirjeldatakse, milliseid funktsionaalsusi võiks veel rakendusele lisada.

### 3.10 Tulevikuplaani võimalused

Autor näeb, et targa toidukaalu rakendusel on palju arenemisvõimalusi nii ärilisi kui ka kasutaja mugavuse parandamiseks.

Tulevikus võiks olla rakenduses sisse logimise funktsioon, et rakendust saaks kasutada mitme seadme peal korraga. Kasutajal võiks olla võimalus sisestada andmeid, mille põhjal saaks näidata soovituslikku kalorsus ning makrotoitainete jaotust.

Targa toidukaalu rakendus võiks teha koostööd mõne toitumisnõustajaga või mõne olemasoleva programmiga, näiteks FitLap, Erik Orgu või FitCat, kes pakuks kasutajale retsepte koos toitumisalase teabega. Retseptid annaksid kogenenud kasutajale ideid ning lihtsustaksid algajate kasutajate elu, sellega et nad saavad valida toitusid, milles on makrotoitained paigas.

Rakenduses võiks olla võimalus lisada toitusid koos toitumisalase teabega. Nii oleks rakenduses kõik kasutatavad toidud olemas ning kasutajad ei tekiks probleeme uute toitute tarbimisega.

Targa toidukaalu rakendusel oleks võimalik ka laieneda teistesse riikidesse. Selleks oleks vajalik lisada rakendusse keelevahetuse funktsionaalsus ning teiste riikide toidud. Teistesse riikidesse arenemisega saaks rakenduses mitmekordistada potentsiaalsete klientide hulka.

Kindlasti saaks ka lisada rakendusse sporditegemise andmete lisamise võimalus, kuid siis juba muutuks rakenduse põhiline eesmärk. Rakendust saaks kindlasti eelnevalt kirjeldatud võimalustega muuta inimestele atraktiivsemaks ning suurendada potentsiaalsete klientide hulka.

### **3.11 Valminud lahendus**

Käesoleva töö raames valmis sinihamba toidukaal ning sellega ühilduv rakendus. Järgnevalt võetakse kokku, miks valminud lahendus on parem turul olevatest toodetest ning millised on rakenduse funktsionaalsused.

Eestis elavad inimesed kasutavad toitumise jälgimiseks peamiselt rakendusi Nutridata ning MyFitnessPal. Mõlemates lahendustes on enamik Eestis müüdavaid toitusid ent kumbki rakendus ei ühildu mõne toidukaaluga ehk kaalunäidu peab sisestama käsitsi. Käesolevas töös valminud rakendus ühildub sinihamba toidukaaluga ning seega on kasutajal valminud lahendust mugavam kasutada.

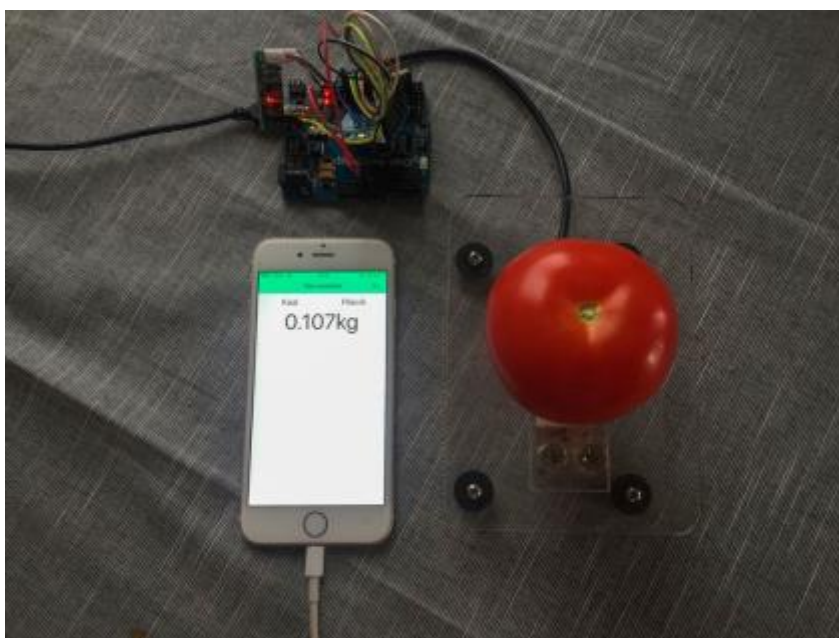
Rakenduses on 4310 erinevat Eestis müüdavat toitu, mis katavad suurema osa Eestis müüdavast toodetest. Kõikides olemasolevates toidukaaluga ühilduvates rakendustes puuduvad paljud Eestis müüdavad toidud. Mistõttu on Eestis elaval inimesel toitumise jälgimiseks mõistlikum kasutada käesolevas töös valminud lahendust.

Valminud mobiilirakenduses on funktsionaalsusi vähem kui algul oli plaanitud. Esialgu oli plaanitud, et rakenduses on sisse logimise süsteem ning kasutaja andmete lisamise võimalus, et kuvada kasutajatele informatsioon sellest, kui mitu kilokalorit nad peaksid tarbima ning kuidas makrotoitained võiksid jaotuda. Töö käigus selgus, et sobivat valmis sinihamba toidukaalu turul ei ole ning seega tuli kaal ise valmistada ning seega ei olnud ajaliselt võimalik kõiki funktsionaalsusi realiseerida. Valminud mobiilirakendus on võimalik ühendada sinihamba signaali abil toidukaaluga ning seejärel kuvatakse kaalunäit mobiilirakenduses, kusjuures ühendamine toimub automaatselt. Mobiilirakenduses saab näha toitude toitumisalast informatsiooni ning toitude sorteerimiseks on valminud otsingumootor. Toitusid on võimalik lisada päevikusse. Päevikus saab vaadata kuupäeva järgi söödud toitude nimekirja koos lisainformatsiooniga. Kusjuures teabe kalorite, valkude,

süsivesikute ja rasvade kogused on arvatud vastavalt kaalunäidule, mis parajasti lisamisel oli. Kõik lisatud funktsionaalsused töötavad valminud lahenduses korrektselt. Valminud lahendus on kuvatud joonistel (vt. Joonis 23 ja Joonis 24).



Joonis 23. Valminud lahendus



Joonis 24. Valminud lahenduse kaalunäide

Kvaliteedi tagamiseks valmistati toidukaal nii, et see suhtleks rakendusega probleemideta. Rakenduse arendamisel kasutati populaarseid ning tunnustatud lahendusi, et tagada rakenduse töökindlus.

Lahenduse valmimisel pidevalt testiti rakendust ning lõpp produkt testiti põhjalikult. Rakenduses testiti kaalunäitu asetades toidukaalule erinevaid objekte ning kontrolliti, kas näit vastab tegelikule objekti kaalule. Seejärel prooviti lisada erinevaid toitusid päevikusse ning kontrolliti, kas väljund vastab tegelikule tulemusele. Lisaks kontrolliti eraldi, kas kalender ning otsingu mootor töötavad vastavalt nõuetele. Kui testimisel ilmus mõni viga, siis puudus likvideeriti ning testiti uuesti.

#### 4. Kokkuvõte

Käesoleva töö esimeseks eesmärgiks oli selgitada tasakaalustatud toitumise olemust ning vajalikkust. Tasakaalustatult toitudes saab organism kätte kõik funktsioneerimiseks vajalikud toitained. Tasakaalustatud toitumise aluseks on makrotoitainete optimaalne jaotus. Organism vajab tasakaalustatult nii rasvasi, valke kui süsivesikuid.

Teiseks eesmärgiks oli kirjeldada Eestis elavate inimeste toitumisharjumusi ning kaaluprobleeme. Tervise Arengu Instituudi uuringust selgus, et koguni 52% täiskasvanutest on ülekaalulised ning 2,4% alakaalulised. Ülekaalulisuse peamisteks põhjusteks on alkoholi liigtarbimine, pidev süsivesiku rikaste toitude tarbimine, üle söömine, vähene liikumine ning harvem haigused. Alakaalulisuse tekke põhjusteks võivad olla vähene toiduenergia tarbimine ja seal juures ka suur treeningkoormus ning harvem haigused.

Kolmandaks uuriti olemasolevaid toidukaaluga ühilduvaid rakendusi. Uuriti kahe rakenduse Situ ja Smart Diet Scale sobilikkust Eestis elava inimese toitumise jälgimiseks ning selgus, et kumbki nimetatutest ei ole sobiv. Peamiseks probleemiks on, et kumbki rakendustest ei sisalda enamikke Eestis müüdavaid toite.

Töö praktiline eesmärk oli valmistada toode, mis lihtsustaks Eestis elava inimese toitumise jälgimist. Bakalaureusetöö raames läbiti tarkvaraarenduse etapid: idee leidmine, visiooni kirjapanek, funktsionaalsete nõuete analüüs, tehniliste lahenduste otsimine (sinihamba toidukaalu otsimine), tulemuse dokumenteerimine ning prototüübi valmistamine. Üheks eesmärgiks oli leida sobiv toidukaal, mida saaks ühendada mobiilirakendusega. Selgus, et turul ei ole ühtegi sobivat sinihamba toidukaalu ning seega valmistati ise nõuetele vastav toidukaal.

Lõputöö raames valminud prototüüp on varajase kontseptsioonilahenduse staadiumis. Rakendusele saaks lisada juurde funktsionaalsusi, mis teeksid kasutaja toitumise jälgimist veelgi lihtsamaks. Samuti oleks rakendusega võimalik laieneda välisturgudele.

Enne lõputöö kirjutamist puudus autoril varasem kogemus riistvara programmeerimise valdkonnas, mistõttu oli sinihamba kaalu valmistamine uudne ning huvitav kogemus. Autoril oli eelnevalt vähe kogemusi iOS operatsioonisüsteemi mobiilirakenduse kirjutamisega, seega paljude funktsionaalsuste realiseerimisele eelnes põhjalik eeltöö. Autori jaoks oli töö kirjutamise kõige õpetlikumaks osaks kaalu valmistamine, kuivõrd ehitamine tundus esialgu keeruline protsess, kuid toote lõplik lahendus pakkus rahulolu ning uhkust. Lõputöö praktilise osa tegemise käigus omandas autor põgusad teadmised

riistvaralise programmeerimise ja ühendamise kohta ning arendas iOS operatsioonisüsteemi mobiilirakenduse kirjutamise oskusi.

## 5. Viidatud kirjandus

- [1] Tekkel M, Veideman T. Eesti täiskasvanud rahvastiku tervisekäitumise uuring, 2016. [https://intra.tai.ee//images/prints/documents/149069399613\\_Eesti\\_taiskasvanud\\_rahvastiku\\_tervisekaitumise\\_uuring\\_2016.pdf](https://intra.tai.ee//images/prints/documents/149069399613_Eesti_taiskasvanud_rahvastiku_tervisekaitumise_uuring_2016.pdf) (12.02.2017)
- [2] Liebert T, Maser M, Pappel K, Pitsi T, Saava M, Sooba E, Vaask S, Vihalemm T, Villa I. Eesti toitumis- ja toidusoovitused. Tallinn. 2006
- [3] Benedict F G, Harris J A. A Biometric Study of Human Basal Metabolism. 1918.
- [4] Kuidas tervislikult toituda. <http://toitumine.ee/kuidas-tervislikult-toituda> (8.02 2017)
- [5] European Health Interview Survey, 2016 <http://ec.europa.eu/eurostat/documents/2995521/7700898/3-20102016-BP-EN.pdf/c26b037b-d5f3-4c05-89c1-00bf0b98d646>
- [6] Kui palju Eestis tarbitakse alkoholi? Statistika <http://alkoinfo.ee/et/moju/alkoholi-toime/ulevaade-statistikast/> (14.01 2017)
- [7] Crawrod N. Definition of Empty Calories. 20 <http://www.livestrong.com/article/475404-definition-of-empty-calories/> (13.02 2017)
- [8] Energia jaotus toidukordade vahel. <http://toitumine.ee/kehakaal/kehakaalu-langetamine/toidusedel/energia-jaotus-toidukordade-vahel> (13.02.2017)
- [9] Itearduino UNO pilt. [http://dvrobot.ru/images/products/product\\_add\\_pic\\_415.jpg](http://dvrobot.ru/images/products/product_add_pic_415.jpg) (8.05.2017)
- [10] Arduino Introduction. <https://www.arduino.cc/en/Guide/Introduction> (9.05.2017)
- [11] Dejan Agostini. Creating a Temperature Sensor for iOS Using BLE and Arduino. 2017. <http://agostini.tech/2017/02/20/creating-a-temperature-sensor-for-ios-using-ble-and-arduino/> (10.05.2017)
- [12] Load Cell Amplifier HX711 Breakout Hookup Guide. <https://learn.sparkfun.com/tutorials/load-cell-amplifier-hx711-breakout-hookup-guide> (10.05.2017)
- [13] JTAppleCalendar. <https://github.com/patchthecode/JTAppleCalendar> (20.04.2017)
- [14] Situ scale koduleht. <http://situscale.com/> (25.04.2017)
- [15] Smart Diet Scale koduleht. <https://www.smartdietscale.com/> (25.04.2017)

[16] Marsic I. Software Engineering. 2012

## **Lisad**

### **I. Litsents**

Mina, **Krisseliine Pärt**,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose

#### **Tark toidukaal,**

mille juhendajad on Alo Peets, Anne Villems ja Taavi Duvin,

- 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
- 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **11.05.2017**