

Tallinna Ülikool

Digitehnoloogiate instituut

Informaatika õppekava

Restorani menüü rakenduse arendamine Androidi platvormile

Bakalaurusetöö

Autor: Alex Neil

Juhendaja: Romil Rõbtšenkov

Autor:, 2017

Juhendaja:, 2017

Instituudi direktor:, 2017

Tallinn 2017

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu töö tulemus ja seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

.....

(kuupäev)

.....

(autor)

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina Alex Neil (sünnikuupäev: _____)

1. annan Tallinna Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Restorani menüü rakenduse arendamine Androidi platvormile“, mille juhendaja on Romil Rõbtšenkov, säilitamiseks ja üldsusele kättesaadavaks tegemiseks Tallinna Ülikooli Akadeemilise Raamatukogu repositooriumis.
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.

Tallinnas, _____

(digitaalne) allkiri ja kuupäev

Sisukord

Sissejuhatus	6
1. Projekti ettevalmistamine	8
1.1. Projekti struktuur	8
1.2. Tahvelarvutite kasutamise analüüs restoranides	9
1.3. Sarnased olemasolevad rakendused	11
1.3.1 DaShef digital restaurant menu	12
1.3.2 eZee eMenu	13
1.3.3 Menulux Tablet Menu	14
1.4. Restorani menüü arendamise ja disainimise juhised	16
1.5. Androidi arendaja juhised	16
2. Rakenduse disain	18
2.1. Arendamiseks kasutatav tarkvara	18
2.2. Nõuded rakendusele	19
2.3. Rakenduse funktsionaalsus	19
2.4. Rakenduse prototüübi vaadete kujundamine	21
3. Arendus	24
3.1. Menüü andmete laadimine rakendusse	24
3.2. Vaadete arendus	25
3.2.1 MainActivity aken	27
3.2.2 Kategooria valiku vaade	27
3.2.3 Toitude valiku vaade	28
3.2.4 Toidu kirjelduse vaade	30
3.2.5 Tellimuse vaade	30
3.2.6 Tellimuse eelvaade	31

3.3.	Keele valiku lahendus	32
3.4.	Kohanduv teksti suurus	32
3.5.	Rakenduse kasutamine kiosk režiimis.....	33
4.	Testimine ja edasiarendus	35
4.1.	Vigade otsimine.....	35
4.2.	Nõuetele vastavus.....	35
4.3.	Testimine klientidega	36
4.4.	Edasiarendamise ideed	39
	Kokkuvõte.....	40
	Kasutatud kirjandus.....	42
	Summary	44

Sissejuhatus

Restoranides on toidu menüüks juba sajandeid kasutatud paberit, kuid tehnilise progressi saavutused pakuvad palju uusi ja põnevaid võimalusi, mis lubavad teha tellimisprotsessi kaasaegsemaks. Üheks sellistest võimalusest on tahvelarvutite kasutamine pabermenüü asemel.

Käesoleva bakalaaurusetöö eesmärgiks on arendada tarkvara, mis asendaks toitlustusettevõtetes pabermenüüd kaasaegsete elektrooniliste vastu ja mis lubaks klientidel sirvida menüüd ja teha tellimusi kasutades tahvelarvuteid, ning kirjeldada sellist arendusprotsessi.

Teema valikut mõjutas tellimus kliendilt, kes tahtis oma restorani innovatiivsemaks teha ja soovis sellist rakendust, mis töötaks tahvelarvutis ja asendaks tema restoranis pabermenüüd elektroonilistega. Tahvelarvuti platvormiks soovis klient Androidi. Lisaks, autorile teadaolevalt, pole restoranides kohapeal elektrooniliste menüüde kasutamine Eestis veel levinud. Sellekohaseid uurimusi või avaldatud statistikat autor ei leidnud.

Töö eesmärgi saavutamiseks viiakse läbi järgnevad etapid:

- tehakse projekti ettevalmistus;
- visandatakse rakenduse disaini;
- loodud disaini põhjal arendatakse rakendus;
- testitakse loodud rakendust;
- tehakse ettepanekud edasisteks arendusteks.

Antud bakalaaurusetöö koosneb viiest osast. Esimeses peatükis kirjeldatakse projekti ettevalmistamise faasi, mille käigus pannakse paika projekti struktuur ja analüüsitakse probleemi tausta. Teine peatükk annab ülevaate rakenduse disaini loomisest. Peatükis selgitatakse välja nõuded rakendusele, kirjeldatakse arendamiseks kasutatavat tarkvara ja rakenduse funktsionaalsust ning räägitakse rakenduse prototüübi kasutajaliidese kujundamisest. Kolmas peatükk on mõeldud lugemiseks eelkõige neile, kellel on Android platvormist mõningad tehnilised teadmised. Selles peatükis räägitakse rakenduse arendamise faasist: kuidas arendati funktsionaalsust ja kasutajaliidest,

tutvustatakse erinevaid probleeme, mis tekkisid arendamise käigus, ja nende lahendusi. Neljandas peatükis antakse ülevaade rakenduse testimisest. Viimases, viiendas peatükis pakutakse erinevaid edasiarendamise ideid.

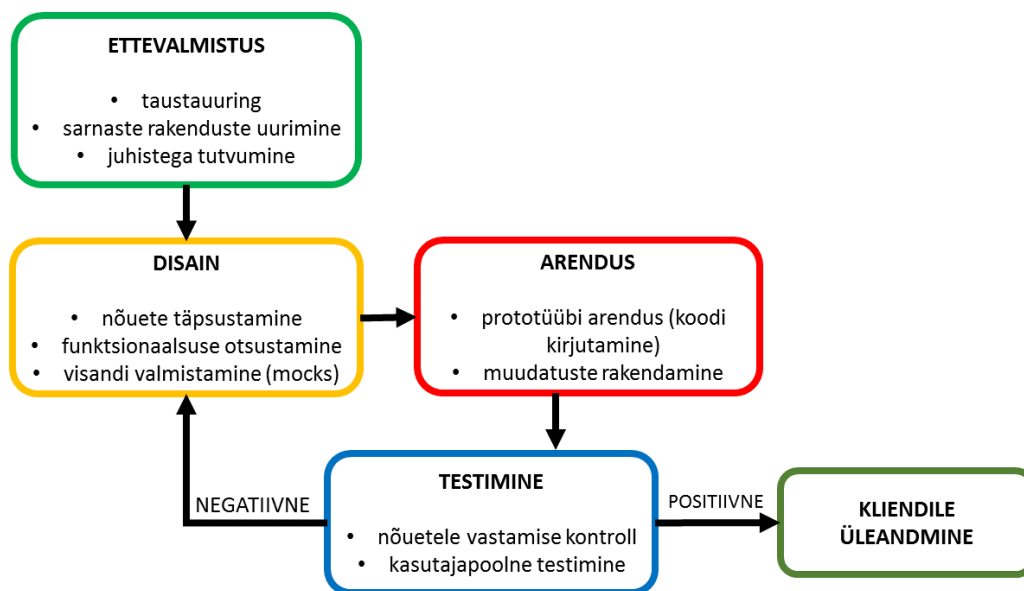
Kuna tegemist on kliendi tellimusel valminud tarkvaraga, siis kliendi õiguste kaitsmiseks rakenduse lähtekoodi antud tööga ei avaldata.

1. Projekti ettevalmistamine

Projekti edukaks läbiviimiseks tuleb kõigepealt otsustada selle struktuur: mida, kuidas ja millises järjekorras teha. Siis tuleb teha taustauuring ja tutvuda juba olemasolevate lahendustega. Nendest sammudest antakse ülevaade järgmistes alapeatükkides.

1.1. Projekti struktuur

Kogu projekti voo võib tingimisi jaotada viieks etapiks: ettevalmistus, disain, arendus, testimine ja valmislahenduse üleandmine kliendile. Arendamisprotsess toimub iteratiivselt. Joonis 1 illustreerib käesolevas töös toimuvat arendusprotsessi.



Joonis 1. Projekti struktuur

Esimene etapp on eeltöö, mille käigus tehakse taustauuring, uuritakse juba olemasolevaid sarnaseid rakendusi ning toimub erinevate juhistega tutvumine.

Järgmiseks etapiks on rakenduse disain, mille käigus selgitatakse välja kliendipoolsed nõudmised rakendusele või kui on tegu uue arendustsükliga, siis toimub nõudmiste täpsustamine. Nõudmiste põhjal otsustatakse, milline peab olema rakenduse funktsionaalsus ning visandatakse rakenduse kasutajaliides.

Arenduse faasi käigus otsitakse tehnoloogilisi lahendusi, mis lubaksid eelmises etapis kavandatud visandi realiseerida, ja kirjutatakse selle realiseerimiseks programmeerimiskood.

Arendusele järgneb testimine. Testimise eesmärk on tuvastada, kas rakendus vastab nõuetele, ja püütakse leida vigu (ingl *bugs*). Testimine toimub esialgu programmeerija (autori) poolt ja kui rakendus jõuab kasutuskõlblikku seisu, siis ka kasutajate (restorani klientide) poolt. Kui testitav rakendus vastab nõuetele ja rakenduse tellija on tulemusega rahul, antakse valmis toode tellijale üle. Vastasel juhul alustatakse uus tsükkel alates disainimisest, kus täpsustatakse nõuded.

1.2. Tahvelarvutite kasutamise analüüs restoranides

Tüüpilises restoranis, kus ei kasutata innovatiivseid tehnoloogiaid, saab toidu tellimise protsessi kirjeldada üldiselt alljärgnevalt (Williams & Julian, 2015):

1. Klient juhatatakse istuma laua taha.
2. Klient jääb ootama ettekandja saabumist, kes toob tall menüü.
3. Peale jookide tellimuse võtmist (tavaliselt) ettekandja lahkub.
4. Klient loeb menüü läbi ja valib toidud, mida soovib tellida.
5. Ettekandja tuleb tellimuse võtmiseks tagasi ning seejärel lahkub ja viib kaasa ka menüüd.

Vaadates sellist tellimuse protsessi, on märgata mitmeid nõrkusi: erinevate etappide vahel on ebamäärase pikkusega ooteajad – näiteks istumaasumise ja ettekandja saabumise vahel (etapid 1 ja 2), samuti menüüga tutvumise ja tellimuse võtmise vahel (etapid 4 ja 5). Traditsioonilise lähenemise veel üks puudustest on see, et pabermenüüdes on toitude informatsioon sageli piiratud või vananenud ja puuduvad toitude pildid. Kasutaja jaoks on sageli keeruline valikute meeldejätmise, kuna tavaliselt pole valikuid võimalik kuskile kirjutada ja kogu tellimus tuleb ettekandjale tellimuse esitamiseks meeles pidada. Üheks puuduseks on ka selline olukord, kus pabermenüül olevat toitu ei ole erinevatel põhjustel siiski võimalik tellida (nt on selleks vajalikud koostisosad lõppenud), kuid pabermenüül vastavate paranduste tegemine on väga keeruline või ei ole mõistlik.

Võttes kasutusele elektroonilised menüüd ja tahvelarvutid, on võimalik parandada tellimuse protsessis mitmeid nõrkusi ning protsessi optimeerida. Näiteks annab elektrooniline menüü informatiivsema ja visuaalsema ülevaate erinevatest toitudest, mis aitab kiirendada ja lihtsustada valikute tegemist. Klient saab soovitud toidud ära

märkida ja esitada tellimuse kohe, kui valikud on tehtud, ning ei pea hakkama uuesti meelde tuletama, mis talle menüü valikutest meeldis.

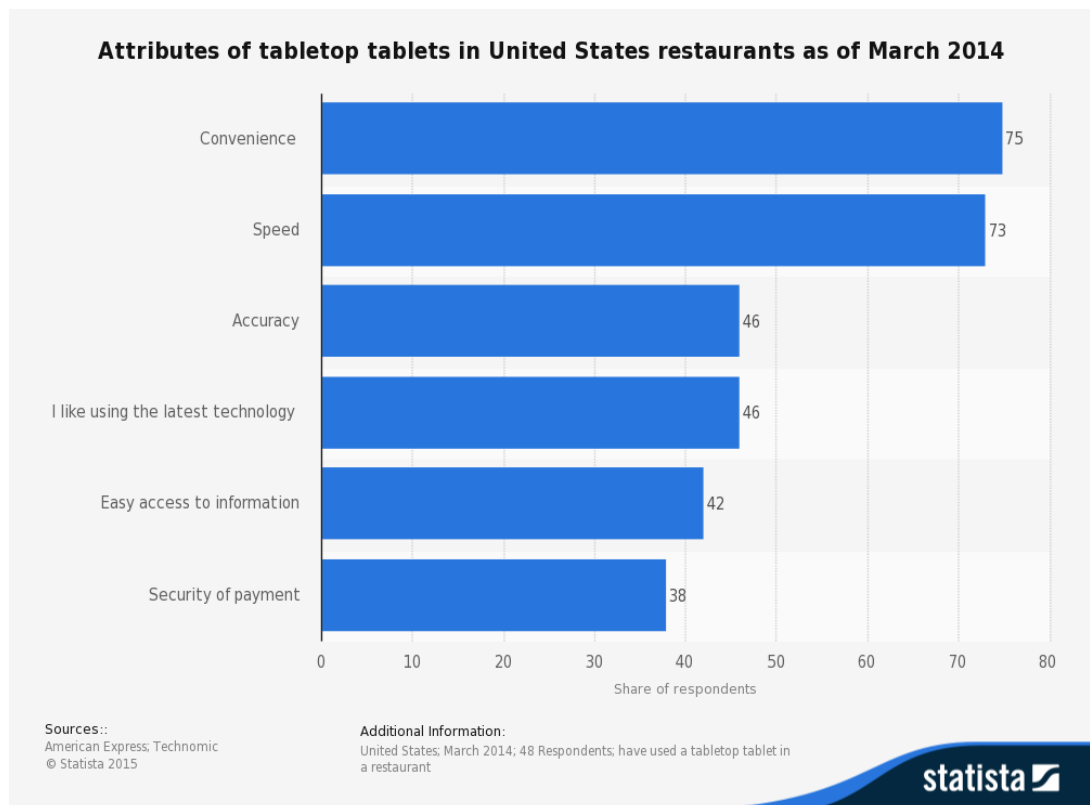
Siin on mõned eelised, mida annab elektrooniline menüü võrreldes pabermenüüga (Williams & Julian, 2015):

- klientidel on parem kontroll tellimisprotsessi üle;
- parem kasutusmugavus;
- optimeeritud tellimisprotsess;
- suurenenud käive tänu optimeeritud tellimusprotsessile;
- näitab firma kaasaegsust ja innovaatsilisust;
- muudatused menüüs ja toitude valikus on lihtsamini tehtavad.

USA-s tehtud uuringus, mille eesmärgiks oli välja selgitada, kas tahvelarvutite kasutamine restoranides soodustab tellimisprotsessi, on välja toodud ka sellise lähenemise puudused, millest olulisemadena võib välja tuua järgmiseid (Hill, 2015):

- Elektroonilise menüü kasutamine võib esialgu vajada aega õppimiseks. Kuni kliendid nendega ära harjuvad, võib ettekandjatel kuluda menüü kasutamise tutvustamisele rohkem aega, mida nad võiksid kasutada pigem tellimuste võtmiseks. Esialgu võib see põhjustada teeninduse aeglustumist.
- Inimliku kontakti kadumine – klientide jaoks võib suhtlus ettekandjatega olla üks osa restorani kogemusest. Tahvelarvutid ei asenda ettekandjaid täielikult, kuid kindlasti vähendavad tegeliku suhtlemise osakaalu.
- Klientidel on keerulisem esitada toitude kohta täpsustavaid küsimusi. Kui elektroonilises menüüs puudub info, mis klienti huvitab, peab ta ikkagi ootama ettekandjat, et saada täpsustavaid vastuseid.

Statista (2014) poolt kogutud statistilised andmed näitavad (Joonis 2), et USA restoranikülastajate jaoks olid tahvelarvutite kasutamisel parimateks omadusteks kasutusmugavus, kiirus ja täpsus. Kasutajates peaaegu pooled (43%) tõid ka välja, et neile meeldib kasutada kaasaegseid lahendusi.



Joonis 2. Tahvelarvutite kasutamise eelised USA restoranides (Statista, 2014)

Toidukohtades tahvelarvutite kasutamise kohta puudub Eestis ülevaade ja ei ole avaldatud statistikat, kuid autorile teadaolevalt on 2017. aastal tahvelarvutid kasutusel vaid üksikutes restoranides.

1.3. Sarnased olemasolevad rakendused

Sarnaste olemasolevate rakenduste otsimiseks kasutas autor Google Play Store¹ teenust, mis on ametlik Google digitaalse media pood. Rakendused, mis on ülevaatesse kaasatud, pidid vastama järgnevatele kriteeriumitele:

- tasuta või tasuta prooviversiooniga rakendused;
- lihtne, atraktiivne kujundus;
- sarnane funktsionaalsus – elektrooniline toidumenüü, mida on võimalik hallata;
- Google Play Store reiting vähemalt neli punkti maksimaalsest viiest.

¹ Google Play Store: <https://play.google.com/store>

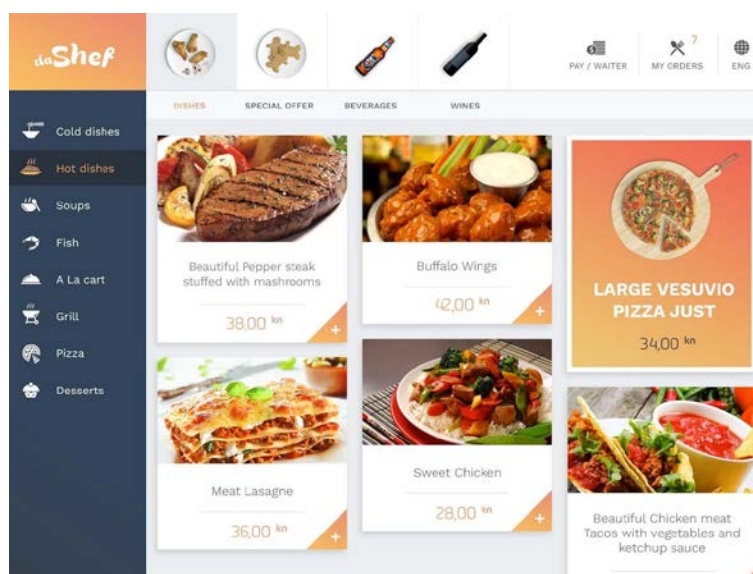
Valituks osutused järgnevad kolm lahendust, mis vastasid ülaltoodud kriteeriumitele: DaShef digital restaurant menu, eZee eMenu, Menulux Tablet Menu.

Uurimise käigus püüdis autor selgitada välja nende rakenduste häid omadusi, mida võiks käesolevas projektis kasutada, ja halbu omadusi, mida tuleks vältida.

Kõik rakendused paigaldati testimiseks tahvelarvutile Samsung Tab 2, mis oli autorile kättesaadav ning piisavalt suure ekraaniga, et rakendusi mugavalt katsetada. Järgmistes alapeatükkides antakse igast rakendusest ülevaade.

1.3.1 DaShef digital restaurant menu

„DaShef digital restaurant menu“² rakendusel (Joonis 3) on olemas tasuta testimisrežiim koos näidismenüüdega.



Joonis 3. DaShef digital restaurant menu

See võimaldab kasutajatel sirvida menüüd ja seda saab kasutada tellimuse esitamiseks ettekandja kaudu või iseteenindusena (tellimuse saatmine rakendusest kassasse). Rakenduse peavaade on jagatud kolmeks sektsiooniks. Ülemises sektsioonis on paneel, mille sakkidelt saab valida toidud, joogid, eripakkumised ning samas asuvad ka nupud keele valiku tegemiseks ja tellimuse vaate kuvamiseks. Peavaate vasakul paneelil on toodete kategooriate valik. Keskmisel, põhipaneelil kuvatakse tooteid kandiliste siltidena koos piltidega ja toote tellimusele lisamise nupuga. Vajutades toote

² DaShef digital restaurant menu: <https://play.google.com/store/apps/details?id=com.dashef.menu>

sildile avaneb hüpikaken koos toote detailsema kirjeldusega ja suurema pildiga. Üllatavalt ei saa sellest hüpikaknast lisada toodet tellimusele. Tellimuse vaates, mille saab avada vastavalt nupult ülemisel paneelil, kuvatakse valikusse lisatud tooted koos fotodega ning toodete muudetavate kogustega. Samuti kuvatakse tellimuse kogumaksumus.

Head omadused:

- erksate värvidega ja kaasaegne visuaalne stiil;
- eraldi on olemas tellimuse ülevaate aken, kus kuvatakse valikusse lisatud tooted;
- loogiline paigutus, kus on kerge orienteeruda.

Puudused:

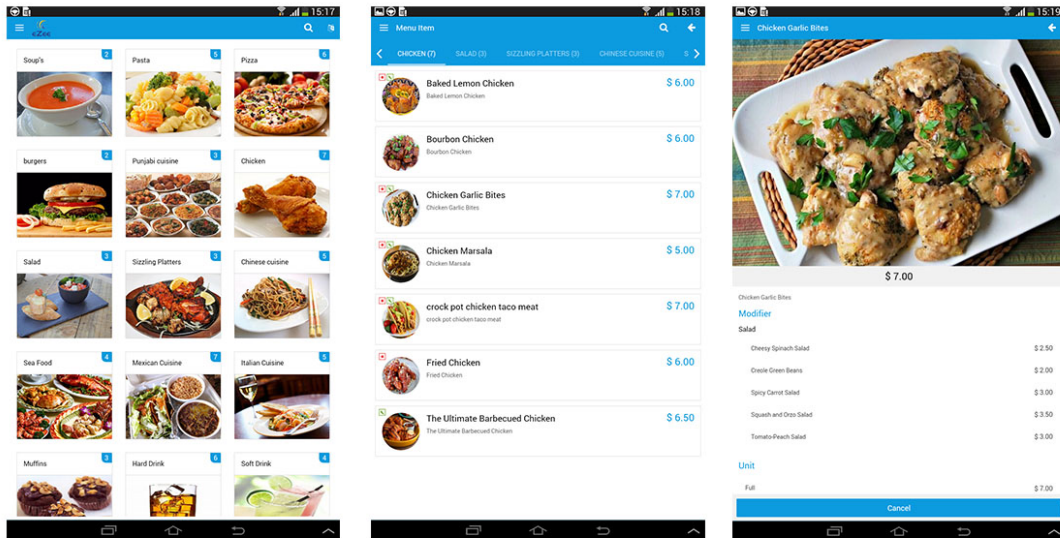
- kasutajaliides toimib aeglaselt, mis on tõenäoliselt põhjustatud liigsest visuaalsete efektide kasutamisest (varjud, läbipaistvus, jms);
- menüü haldamine ja uuendamine toimud läbi arendaja veebiteenuse ja nõuab teenusega liitumist. Kui rakenduse arendaja peaks otsustama lõpetada teenuse toetamise kaotab klient täielikult ligipääsu menüü haldamisele;
- hindu kuvatakse väikses kirjas, mida võib olla keeruline lugeda;
- toote kirjelduse hüpikaknas puudub võimalus toodete lisamiseks tellimusele.

1.3.2 eZee eMenu

„eZee eMenu“³ on tasuta rakendus, mis kuvab restorani menüü kategooriaid, tooteid koos piltide ja hindadega (Joonis 4). Funktsionaalsuse poolest on see väga sarnane pabermenüüle, kuna see vaid kuvab erinevaid tooteid ning sellel puudub toodete valimise funktsioon. Sellest tulenevalt pole kasutajal võimalik huvipakkuvate toodete üle järke pidada ning valikud tuleb endal meelde jätta. Rakenduse avades kuvatakse toitade kategooriad ruudustikpaigutusena (ingl *grid layout*). Kategooriale vajutades avaneb vastava kategooria toitade nimekirja vaade, kus toidud on kuvatud kandiliste siltidena. Igal toote sildil on toote nimetus ja hind. Vajutades toote sildile avaneb toote

³ eZee eMenu: <https://play.google.com/store/apps/details?id=com.ezeetechnosys.ezeemenu>

täieliku kirjelduse vaade koos toote suure foto, täispika kirjelduse ja hinnaga. Rakendus on lukustatud vertikaalsesse paigutusse.



Joonis 4. eZee eMenu ekraanitõmmised

Head omadused:

- kasutajaliides toimib kiiresti, kuna ei kasutata visuaalseid eriefekte, nagu varjusid ja animatsioone;
- lihtne paigutus, kus on kerge orienteeruda;
- meeldiv visuaalne kujundus, mis ei väsita silmi.

Puudused:

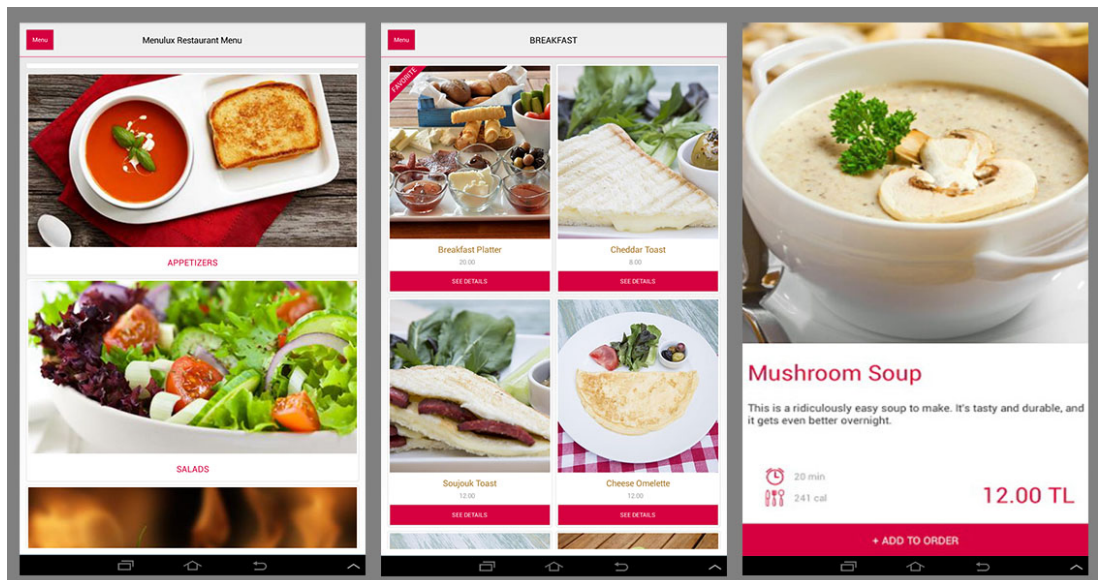
- tasuta versioonis puudub tellimuse koostamise funktsioon;
- pole võimalik tooteid märkida kergeks meeldejätmiseks;
- ei võimalda saada ülevaadet tellimuse koguhinnast.

1.3.3 Menulux Tablet Menu

„Menulux Tablet Menu“⁴ rakenduse (Joonis 5) peavaade kuvab toitude kategooriaid vertikaalse keritava nimekirjana. Iga kategooria on kuvatud suure ekraanilaiuse sildina koos pildiga. Kategoorial vajutades avaneb toodete vaade, kus tooted on kuvatud kahetulbalises paigutuses. Igal toote sildil on kuvatud toote nimetus, pilt, hind ja

⁴ Menulux Tablet Menu: <https://play.google.com/store/apps/details?id=com.menulux.menu>

tellimusele lisamise nupp. Toidu sildile vajutades kuvatakse kasutajale toote detailse kirjelduse vaade, kus on toote suur foto, nimetus, täispikk kirjeldus, hind ja ka tellimusele lisamise nupp. Kogu tellimuse nägemiseks peab kasutaja vajutama “My order” nupule, mis paikneb igas vaates üleval menüüribal. Rakendus on lukustatud vertikaalsesse paigutusse.



Joonis 5. Menulux Tablet Menu ekraanitõmmised

Head omadused:

- lihtne visuaalne stiil;
- kasutajaliidese kiire toimimine;
- suured pildid.

Puudused:

- kuna kategooriaid kuvatakse suurte ekraanilaiuste siltidena, on keeruline saada ülevaadet kõikidest toodete kategooriatest korraga – 10-tollise ekraaniga tahvelarvutil oli korraga nähtaval vaid 2,5 kategooriat;
- sama probleem on ka toitude valiku vaates – 10-tollise ekraaniga tahvelarvutil on korraga kuvatud vaid 4 toodet;
- teksti suurus on liiga väike ja keeruline lugeda;
- vaadete vahel liikumiseks puudub tagasiliikumise nupp – ainus võimalus tagasi liikumiseks on tahvelarvuti enda tagasinupp.

1.4. Restorani menüü arendamise ja disainimise juhised

Menüü on restorani jaoks kõige olulisem turunduse ja müügi vahend ning mõjutab otseselt, mida kliendid tellivad ja kui palju nad kulutavad, mistõttu mõjutab menüü disain otseselt restorani käivet (Pavesic, kuupäev puudub).

Menüü rakenduse paremaks disainimiseks analüüsi autor erinevaid juhiseid (Creger, 2014; Gentile, 2014; Pavesic, kuupäev puudub) ja valis nendest alljärgnevad punktid, mida autor pidas kõige olulisemateks:

- Oluline on värvide valik. Menüü disain peab olema kooskõlas restorani/firma üldise värvilahenduse ja kujundusega.
- Menüü peab olema loogiliselt struktureeritud.
- Trükitähti peab kasutama ettevaatusega. Toitude nimetused võib kirjutada trükitähtedega, kuid toidu kirjeldus ja muud elemendid, kuhu soovitakse suunata kliendi tähelepanu, peaksid olema tavalises kirjas – see aeglustab lugemise kiirust, et klient süveneks teksti rohkem.
- Fondi suurus, värv ja tüüp peavad olema valitud selliselt, et menüü tekst oleks kergesti loetav.
- Elementide esiletõstmist kastide või raamidega võib kasutada sisu rõhutamiseks või tähelepanu püüdmiseks.

1.5. Androidi arendaja juhised

Järgnevalt tutvustatakse üldiseid põhimõtteid Android platvormil arendamiseks. Ülevaade on tehtud Androidi arendaja ametliku juhise (Google Inc, kuupäev puudub) põhjal.

Androidi rakenduste arendamise ametlik programmeerimiskeel on Java. Android SDK (tarkvaraarenduskomplekt) töövahendid kompileerivad koodi koos andmete ja algfailidega Androidi paketi (APK), mis on tegelikult arhiivifail .apk laiendiga. Üks APK fail sisaldab kogu Androidi rakenduse sisu ja on see fail, mida kasutatakse rakenduse installeerimiseks Android seadmele.

Androidi rakendus sisaldab rohkem kui ainult koodi – see nõuab ka teisi ressursse, mis on koodist eraldi, nagu pildid, audiofailid ja muu, mis on vajalik rakenduse visuaalseks kujunduseks. Animatsioonid, menüüd, stiilid, värvid ja kasutajaliidese paigutus

defineeritakse XML failides. Kasutades eraldi olevaid ressursse on lihtsam uuendada rakenduse erinevaid näitajaid, kuna alati ei pea uuenduseks koodi muutma.

Androidi platvormile rakenduse arendamine on sarnane teistele platvormidele arendamisega. Järgnev nimekiri annab ülevaate protsessist, kuidas arendada Androidi rakendust kasutades Android Studio⁵-t (Google Inc, kuupäev puudub):

1. Töökeskkonna seadmine. Selle etapiga paigaldatakse Android Studio ja luuakse uus projekt.
2. Rakenduse kirjutamine – programmeerimiskoodi kirjutamine, kasutajaliidese disainimine ja vajalike ressursside loomine.
3. Rakenduse ehitamine (ingl *build*) ja jooksutamine. Selles faasis on rakendus arendatud vigade tuvastamist võimaldavaks (ingl *debuggable*) APK paketiks, mida saab paigaldada ja jooksutada emulaatoril või Androidiga seadmel.
4. Vigade otsimine (ingl *debug*) ja testimine. See on iteratiivne faas, kus jätkatakse programmi kirjutamist, kuid tähelepanu on suunatud vigade kõrvaldamisele ja jõudluse optimeerimisele.
5. Avaldamine. Kui rakenduse testimine kinnitab, et see rakendus toimib vastavalt plaanitud disainile, antakse rakendusele versiooni number ja see avaldatakse kasutamiseks.

⁵ Android Studio: <https://developer.android.com/studio/index.html>

2. Rakenduse disain

Rakenduse kavandamisel tuleb otsustada, milliseid vahendeid kasutada rakenduse arendamiseks, ning välja selgitada kliendipoolsed nõudmised rakendusele. Nõudmiste põhjal tuleb otsustada, milline peab olema rakenduse funktsionaalsus. Kui funktsionaalsus on otsustatud, hakkab rakenduse kasutajaliidese visandi disainimine. Lisaks kliendi nõudmiste arvestamisele toetutakse rakenduse arendamisel ka projekti ettevalmistamisel tehtud olemasolevate rakenduste ülevaatele (ptk 1.3) ning juhiste (ptk 1.4).

2.1. Arendamiseks kasutatav tarkvara

Androidi rakenduste arenduseks on kasutusel kaks suuremat Java programmeerimiskeelel põhinevat integreeritud programmeerimiskeskonda (ingl *Integrated Development Environment, IDE*) – Android Studio ja Eclipse⁶. Antud projekti jaoks valiti arendusplatvormiks Android Studio, millel on Eclipse ees mitmeid eeliseid (Mehul, 2015):

- see põhineb väga populaarsel integreeritud programmeerimiskeskonnal IntelliJ IDEA arendajalt JetBrains ning seda pakub Google tasuta;
- parem automaatne koodi soovitus (ingl *Advanced Code Completion*);
- pukseerimine (ingl *drag-and-drop*) tugi arendamiseks graafilist kasutajaliidest;
- Androidi emulaator, millel on suurem töökiirus kui enamikul Androidi seadmel, tänu millele on arendusprotsess kiirem.

Graafiliste elementide loomiseks kasutatakse tarkvara Adobe Photoshop, millest on saanud *de facto* standard rastergraafika töötlemisel.

Versioonihalduseks kasutab autor GitHub veebikeskonda, mis põhineb Git hajutatud versioonihaldustarkvaral. Versioonihaldus lubab failidesse tehtud muudatusi jälgida, võrrelda ja vajadusel muudatusi tagasi võtta. GitHub on Android Studio-ga hästi integreeritud.

⁶ Eclipse: <https://www.eclipse.org/>

2.2. Nõuded rakendusele

Plaanitava rakenduse funktsionaalsete ja visuaalsete nõudmiste väljaselgitamiseks toimus kohtumine kliendiga. Ühise arutelu tulemusel otsustati, et rakendus peab vastama järgmistele nõuetele:

- rakendus peab olema optimeeritud kasutamaks GoClever Quantum 2 tahvelarvutitel (10-tolline ekraan, 1024x600 ekraani resolutsioon, Android 5.1.1), mida klient oli eelnevalt hankinud;
- rakendus peab töötama kahes keeles: eesti ja vene;
- toidud peavad olema organiseeritud kategooriatesse;
- klient saab navigeerida toitude kategooriates;
- iga toode peab sisaldama fotot, kirjeldust ja hinda;
- kliendil peab olema võimalus valikuid tellimiseks salvestada;
- kliendil peab olema võimalus menüüs tehtud valikuid muuta;
- peab olema valik kogu tellimuse kiireks kustutamiseks;
- klient peab saama tooteid valikule lisada ja muuta koguseid;
- rakendus peab toimima ilma võrguühenduseta ning rakenduse kõik andmed peavad olema salvestatud tahvelarvutisse lokaalselt;
- menüü peab olema kergelt hallatav ja uuendatav;
- kasutajaliidese disainile ja värvilahendusele pole kindlaid nõudmisi, kuid kasutajaliides peab toimima sujuvalt;
- heliefektid pole vajalikud;
- rakendust saab käivitada kiosk-režiimis (tahvelarvuti on lukustatud ainult rakenduse kasutamiseks ja kasutaja ei pääse ligi tahvelarvuti muudele rakendustele).

2.3. Rakenduse funktsionaalsus

Väljaselgitatud nõuete põhjal otsustati, milline peab olema arendatava rakenduse funktsionaalsus.

Loodava elektroonilise menüü rakenduse põhiline eesmärk on anda restorani kliendile võimalus toidu menüüd sirvida, tooteid tellimusele lisada ja eemalda ning annab ülevaate tellimusest ja selle maksumusest.

Rakenduse ekraan on jagatud seksioonideks: üleval on rakenduse menüüriba, kus kuvatakse restorani nimi või mõni muu informatsioon sõltuvalt kontekstist, ning nupud eesti ja vene keele vahel vahetamiseks. Alumises seksioonis on ülevaade tellimusest, kus valitud tooteid kuvatakse eelvaate piltidena koos koguse infoga, tellimuse kustutamise nupp ja tellimuse akna avamise nupp. Keskmise seksiooni katab ekraanist suurema osa – selle sisu on muutuv ja näitab erinevaid vaateid. Nii ülemine kui ka alumine seksioon on alati nähtavad, keskmise seksiooni sisu muutub vastavalt kliendi valikule. Menüüriba tekst muutub vastavalt keskmise seksiooni sisule. Rakenduse keskmises seksioonis kuvatakse ühte järgmisest vaadetest:

- toidukategooria vaade;
- toidu valiku vaade;
- konkreetse toidu täiskirjelduse vaade;
- tellimuse vaade.

Toidukategooria vaates kuvatakse toidu kategooriaid ruudukujuliste siltidena koos piltide ja nimetustega. Kui kategooriaid on rohkem kui ekraanile mahub, võib vaadet kerida. Kategooria sildile vajutades avaneb uus vaade – toidu valiku vaade koos valitud kategooriasse kuuluvate toitudega. Menüüribal kuvatakse restorani nimi.

Toidu valiku vaates võib näha kõiki valitud kategooriasse kuuluvaid toite. Toidud kuvatakse vertikaalse nimekirjana, mis koosneb ekraanilaiustest siltidest. Kui silte on rohkem kui ekraanile mahub, võib nimekirja kerida. Igal sildil on toidu pilt, nimi, lühikirjeldus, hind ja nupud toidu tellimusele lisamiseks ja tellimusest kustutamiseks. Toidu sildile vajutades avaneb toidu täieliku kirjelduse vaade. Kui toidu valiku vaade on aktiivne kuvatakse menüüribal ekraanil olevate toitude kategooria.

Toidu täieliku kirjelduse vaates võib näha toidu nime, suurt toidu pilti, täiskirjeldust, hinda ja tellimusele lisamise ja tellimusest eemaldamise nuppe. Menüüribal kuvatakse toidu kategooria.

Tellimuse vaates näeb klient tellimusele lisatud toitusid. Toite kuvatakse vertikaalselt keritava nimekirjana, kus iga toit on kujutatud sildina täpselt nagu ka toidu valiku vaates, kuid millel on ka valitud toitude kogused. Kui klient vaatab tellimuse vaadet,





muutub alumise sektsiooni sisu – tellimuse ülevaate asemel kuvatakse seal tellimuse koguhind. Menüüribal kuvatakse tekst „Teie tellimus“.

Nõuete järgi peab rakendus võimaldama lihtsat lokaalset (ilma Wi-Fi ja internetita) sisu haldust. See tähendab, et kõik menüüga seotud andmed (tekstid ja pildid) peavad olema salvestatud tahvelarvuti sisemisel mälul (ingl *internal storage*) eraldi kaustas.

2.4. Rakenduse prototüübi vaadete kujundamine

Vastavalt eelnevalt kirjeldatud funktsionaalsusele (ptk 2.3) ja võttes arvesse varem vaadeldud menüü kujundamise juhendeid (ptk 1.4) ning teisi sarnaste rakenduste häid omadusi (ptk 1.3), koostatakse rakenduse erinevate vaadete visandid. Kokku valmistatakse 4 visandit.

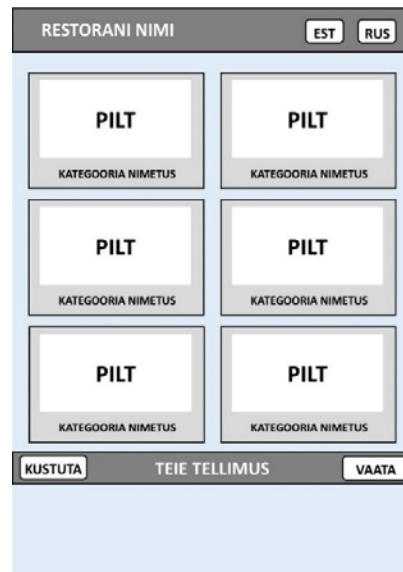
Eelnevalt uuritud juhise soovitus järgi (Creger, 2014) oli otsustatud programmi lõppversioonis kasutada rakenduse tellija restorani veebilehe stiili (Joonis 6): hallid taustad, musta ja punast värvi teksti, tumedad pildid, kandilised kujunduselemendid.

Supid		Hind
	#137 Juurvilja Supp šampinjonid, porrulauk, rohelised oad, paprika, seesamiseemned, hiina kapsas.	3,50 €
	#84 UUS! Ebi supp tiigerkrevetid, juurviljad, kimchi kaste	5,00 €
	#85 UUS! Kreemsupp lõhefileega lõhefilee, juurviljad, koor	5,00 €
	#86 Miso supp Merirohi, tofu, seesamiseemned	2,80 €

Joonis 6. Näide rakenduse tellija veebilehe kujundusest

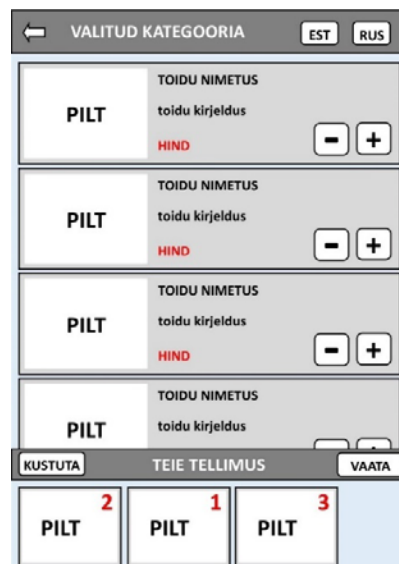
Toidu kategooriate valiku aknas (Joonis 7) saab klient näha kõiki toitude kategooriaid, vahetada keeli ja näha lühiülevaadet oma tellimusest. Keelevahetuse

võimalus säilib kõigis teiste vaadetes. Toidu kategooria valiku aken on samaaegselt ka rakenduse avavaateks.



Joonis 7. Visand toidu kategooriate vaatest

Toidu valiku vaates näeb klient kõiki toitusid, mis kuuluvad eelnevas sammus valitud kategooriasse, ja lühiülevaadet oma tellimusest (Joonis 8). Vajutades pluss või miinus nuppudele saab toitu tellimusele lisada või tellimusest kustutada. Vajutades toidu sildile avaneb toidu täiskirjelduse aken. Menüüribale kuvatakse tagasiliikumise nool.



Joonis 8. Toidu valiku vaate visand

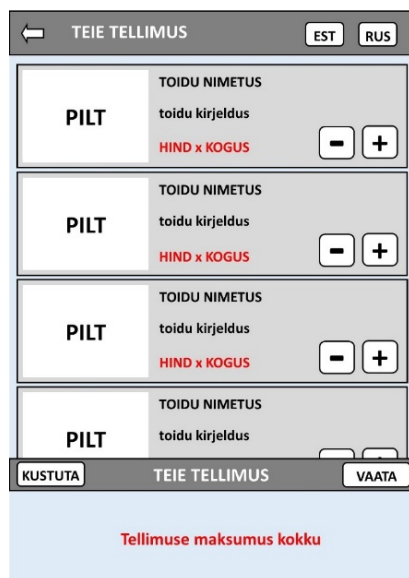
Toidu kirjelduse vaates näeb klient toidu suuremat pilti, täiskirjeldust koos toidu hinnaga ning ülevaadet oma tellimusest (Joonis 9). Vajutades pluss või miinus

nuppudele saab toitu tellimusele lisada või tellimusest kustutada. Menüüribal on kuvatud tagasiliikumise nool.



Joonis 9. Toidu kirjelduse vaate visand

Tellimuse vaates võib näha kõiki tellimusele lisatud toitusid koos piltide ja hindadega ning alumises osas kuvatakse tellimuse kogumaksumus (Joonis 10). Vajutades pluss või miinus nuppudele saab toidu kogust muuta. Menüüribal on kuvatud tagasiliikumise nool.



Joonis 10. Tellimuse vaate visand

3. Arendus

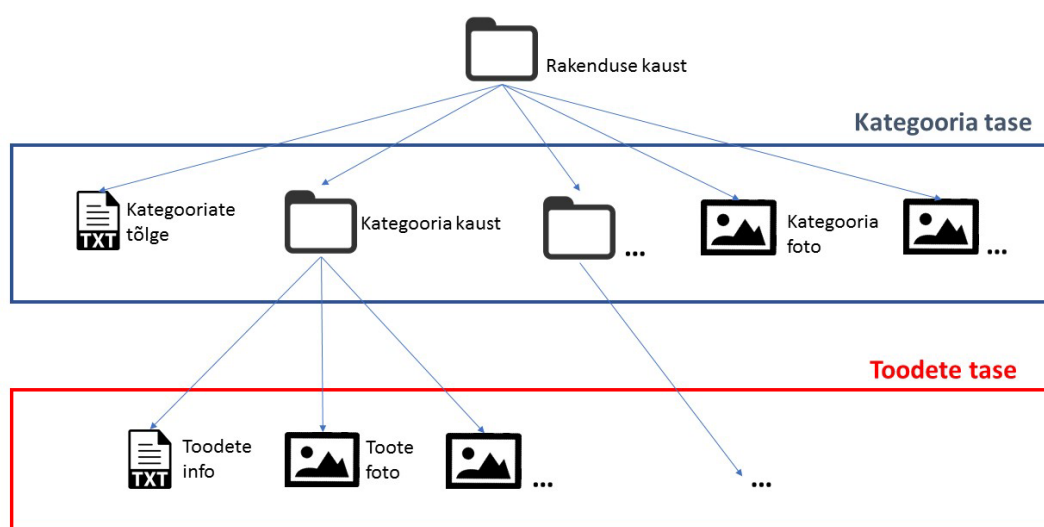
Järgmiseks etapiks on rakenduse funktsionaalsuse programmeerimine, lähtudes loodud kasutajaliidese visanditest ja toetudes varasemas etapis väljaselgitatud nõuetele ning vaadeldud juhistele. Selles peatükis kirjeldatakse arenduse käiku, selle käigus tekkinud põhilisi probleeme ja nende lahendusi.

Arendusplatvormiks on Android Studio versioon 2.2.3 ning rakendus kirjutatakse kasutades Java programmeerimiskeelt. Versioonihalduseks kasutatakse GitHub veebikeskkonda.

Käesoleva töö parema loetavuse tagamiseks tähistatakse **eristiiliga** kõik koodi lõigud ja klassid, mis on loodud antud projekti raames arendatava menüü rakenduse jaoks ja mis ei kuulu Androidi standardsesse teeki.

3.1. Menüü andmete laadimine rakendusse

Kuna klient soovis, et tahvelarvutid töötaksid võrguühenduseta, arendati rakenduse jaoks järgnev toitudega seotud andmete (nt fotod, nimetused, kirjeldused jne) haldamise ja rakendusse laadimise süsteem. Selle kohaselt salvestatakse andmed määratud kausta tahvelarvuti sisemälus – sel viisil on rakenduse tellijal edaspidi lihtsam andmeid hallata, kuna selline haldus ei nõua erilisi IT-alaseid oskusi. Andmed on loogiliselt struktureeritud failisüsteemi tasemel (Joonis 11).



Joonis 11. Andmete struktuur failisüsteemi tasemel

Rakenduse peakaustas on iga kategooria jaoks enda alamkaust vastava kategooria nimega. Samuti on rakenduse peakaustas kategooriate pildid vastavate kategooriate nimedega. Lisaks on selles kaustas tekstifail kategooriate venekeelsete tõlgetega. Iga kategooria kausta sees on vastava kategooria toitude fotod ja tekstifail, kus on kogu info vastavate toitude kohta kindlalt struktureeritud ja koolonitega eraldatud tekstina: nimetus eesti keeles, kirjeldus, hind, foto faili nimi, venekeelne nimetus ja venekeelne kirjeldus. Kui venekeelne tõlge on puudu, kasutatakse eestikeelset vastet. Näiteks üks andmerida tekstifailist: `Miso supp : Merirohi, tofu, seesamiseemned : 2,80 € : miso_supp.jpg : Суп Мисо : Морская трава, тоффу, кунжут.`

Andmete kausta saab ette valmistada arvutis ja kogu kausta saab kergelt kopeerida USB kaabli kaudu ühendatud tahvelarvutisse või jagada välise mälukaardi abil.

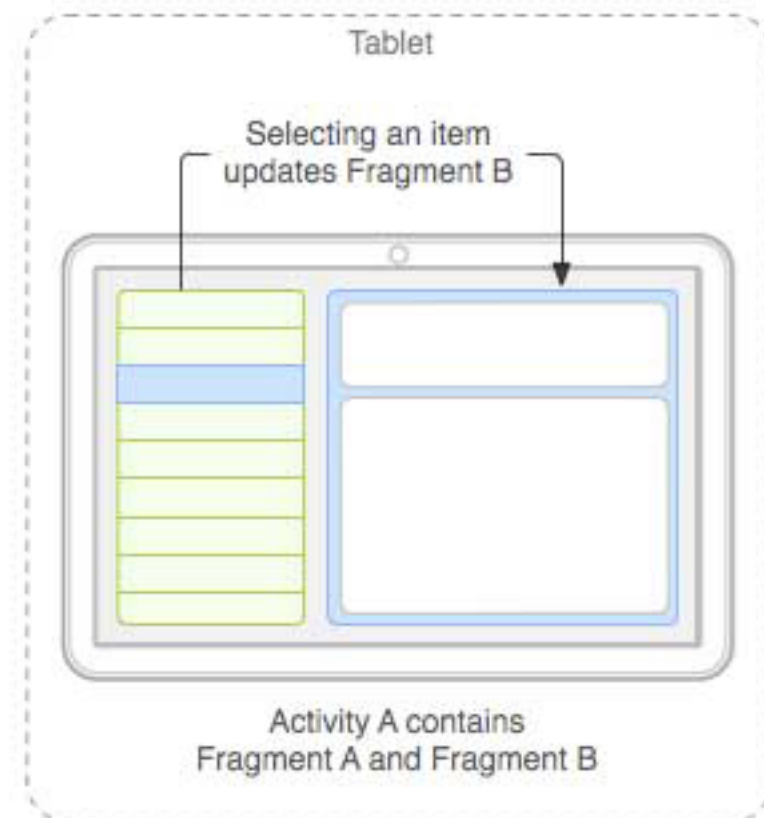
Rakenduse avamisel skaneeritakse määratud kausta sisu ja saadakse kategooriate nimed, fotode failiteed (ingl *path*) ja toodete andmed. Sisemiselt salvestab rakendus andmed Java *List* andmestruktuuri, näiteks eestikeelsed kategooriate nimed salvestatakse *List*-i nimega `categoriesNamesEst`. Java *List* andmestruktuur on kergesti hallatav ja mugav vahend struktureeritud andmete hoidmiseks. Toodetega seotud andmete hoidmiseks luuakse *List*, kus iga toode on esindatud ühe `FoodItem` klassi objektina. `FoodItem` on selle rakenduse jaoks spetsiaalselt loodud klass, kus hoitakse toodetega seotud andmeid, ja millel on järgmised väljad: nimetus, kirjeldus, hind, jne.

3.2. Vaadete arendus

Ühe kasutajaliidesega akna nimetuseks Androidi süsteemis on *activity*. Ühel rakendusel saab olla üks või mitu *activity*-t. *Activity* klass vastutab rakenduse akna loomise eest ja kõikide kasutajaliidese komponentide laadimise eest kasutades XML faili, mis on salvestatud projekti *res/layout* kausta (Google Inc, kuupäev puudub).

Tüüpiline viis Android Studio-s rakenduse loomiseks, on esialgu luua *activity* paigutus kasutades Android Studio poolt pakutud pukseerimise (ingl *drag-and-drop*) liidest, ning seejärel kirjutatakse Java kood, et lisada loodud paigutuselementidele funktsionaalsus.

Kuna disainimise etapis oli otsustatud, et mõeldes kasutaja mugavusele peab kasutajal alati olema kõikides vaadetes kiire ülevaade oma tellimusest, otsustas autor kasutada üksikut *activity*-t fragmentidega. Androidi *Fragment* klass kapseldab kasutajaliidese komponendid ja *activity* käitumised moodulitesse, mida saab *activity*-sse kaasata, vahetada ja sealt eemaldada. Fragment käitub mõneti nagu alam-*activity*, millel on oma paigutus ja mis haldab enda elutsükli. Fragmentidega lähenemisviisi eeliseks selle töö käigus arendatavas rakenduses on see, et rakenduse akna saab jagada sektsioonideks, millest on igäüks eraldi hallatav. *Activity* paigutus on jagatud kaheks sektsiooniks, millest kumbki on esindatud fragmentina. Alumise sektsiooni fragmenti kasutatakse erinevates vaadetes, keskmises sektsioonis lisatakse ja eemaldatakse fragmente dünaamiliselt. Sellise lähenemise lisa eelis on see, et andmete vahetamine fragmentide vahel on lihtne, kuna nad on kõik sama *activity* sees. Näiteks valides toote ühes fragmendis uueneb kuvatav sisu teises fragmendis (Joonis 12).

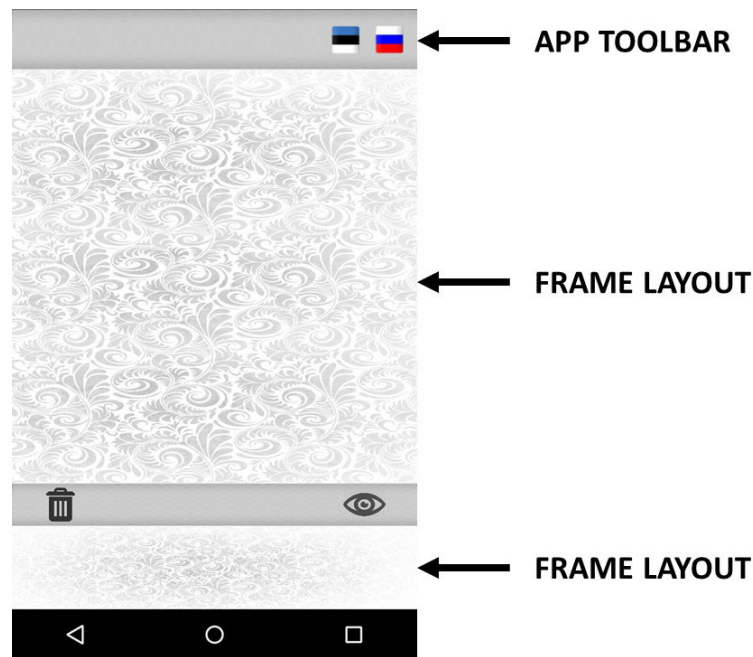


Joonis 12. Fragmentide vaheline interaktsioon (Google Inc, kuupäev puudub)

3.2.1 MainActivity aken

`MainActivity` on ainus rakenduses kasutatav *activity*. See toimib kui tugiraamistik, millele lisatakse erinevaid fragmente. Seda kasutatakse funktsionaalsuseks, mida on tarvis kogu rakenduse piires sõltumata ekraanil kuvatud fragmendist.

See sisaldab ülemist menüüriba tekstiväljaga, mis muutub vastavalt fragmendile, ja kahte nuppu keelte vahel valimiseks. `MainActivity` aknas on kaks *FrameLayout* paigutust, mida kasutatakse erinevate fragmentide kuvamiseks. Kaks fragmenti on eraldatud ribaga, millel on kujutatud prügikasti ikoon tellimuse kustutamiseks ja silma ikoon tellimuse vaatamiseks. `MainActivity` paigutus on näidatud alloleval pildil (Joonis 13).



Joonis 13. MainActivity paigutus

3.2.2 Kategooria valiku vaade

Kategooria valiku fragmendis kasutatakse *GridView* vidinat. *GridView* kuvab elemente kahedimensionaalses keritavas ruudustikus. Ruudustiku elemendid lisab automaatselt *ListAdapter*, mis ühendab omavahel *GridView* ja sellega seotud andmed. *ListAdapter* on *Adapter* liidese alamklass. Android platvormil võimaldab *Adapter* objekt ligipääsu andmetele ja toimib sillana andmete ja vaadete vahel. *Adapter*

vastutab ka *View* vidina loomise eest andmekogu iga elemendi jaoks (Google Inc, kuupäev puudub). *View* elemendi jaoks otsustati kasutada *CardView* vidinat, mis kujutab endast raamikujulist ümarate nurkadega paigutust, millel võib olla ka taust ja vari. Toidu kategoorial vajutades käivitatakse toitude valiku fragment (kategooriate valiku fragment asendatakse toitude valiku fragmendiga). Lõpptulemus on kujutatud alloleval pildil (Joonis 14):



Joonis 14. Kategooria valiku vaade

3.2.3 Toitude valiku vaade

Toitude valiku fragmendi jaoks kasutatakse *ListView* vidinat. *ListView* klass on võimeline kuvama keritavat elementide nimekirja. Nimekirja sisendiks võivad olla suvalised Java objektid. *Adapter* ekstraheerib andmeobjektidest korrektsed andmed ja seob need andmed õigete *View* vidinatega igas *ListView* reas (Lars, 2016). Iga andmeid kuvav *ListView* rida sisaldab paigutust, millel võib olla suvaline keerukus. Siin arendatavas restorani menüü rakenduses sisaldab loendi rida järgmisi elemente: vasakule joondatud pilt, mitmerealine tekst toote nime, kirjelduse ja hinnaga, samuti kaks nuppu toote lisamiseks tellimusele või tellimusest eemaldamiseks (Joonis 15):



Joonis 15. Loendi rea paigutus

Et tuvastada vajutusi loendi ridadel ja nendele vastavalt reageerida, lisatakse *ListView* vidinale *OnItemClickListener* (elemendi vajutuse kuulaja) (Koodinäide 1).

```

listview.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        someAction();
    }
});

```

Koodinäide 1. Listener-i lisamine *ListView* vidinale

Arendatavas rakenduses saab *OnItemClickListener* rea vajutamisel vajutatud elemendi ID ja käivitab vastava ID-ga toidu kirjelduse fragmendi. Toidu valiku vaate tulemus on näidatud alloleval pildil (Joonis 16):



Joonis 16. Toitude valiku vaade

3.2.4 Toidu kirjelduse vaade

Toidu kirjelduse fragmendil on väga lihtne paigutus ja funktsionaalsus (Joonis 17). See sisaldab kolme *TextView* vidinat näitamaks toidu nimetust, kirjeldust ja hinda. Samuti on olemas *ImageView* vidin, mis kuvab toidu fotot kattes kogu paigutusest peaaegu poole. Lisaks on kaks nuppu, millega saab toodet tellimusele lisada või sealt eemaldada.

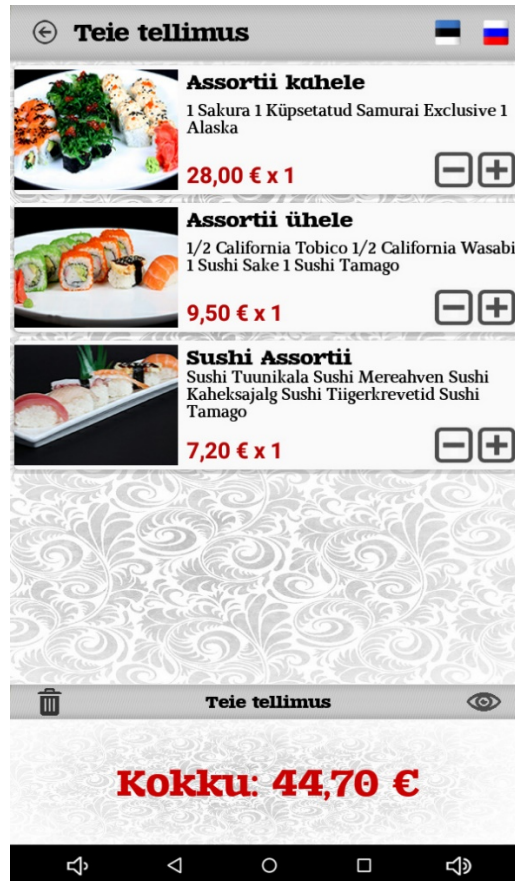


Joonis 17. Toidu kirjelduse vaade

3.2.5 Tellimuse vaade

Tellimuse fragment kasutab samu visuaalseid ja funktsionaalseid lahendusi, nagu toidu valiku fragment (Joonis 18). Ainus erinevus on selles, et see kuvab ainult neid tooteid, mis on eelnevalt tellimusele lisatud. Tellimuse andmete hoidmiseks on loodud spetsiaalne `Order` klass, mis sisaldab Java *List* andmestruktuuri tellimusse lisatud toodete andmetega. `Order` klassil on *public* meetodid `FoodItem` objektide loendile lisamiseks ja sealt eemaldamiseks ning meetod tellimuse kogumaksumuse arvutamiseks.

Kui tellimuse vaate fragment on aktiivne, muutub ekraani alumises osas paikneva tellimuse eelvaate fragmendi sisu – tellimuse eelvaate asemel kuvatakse seal tellimuse kogumaksumust.

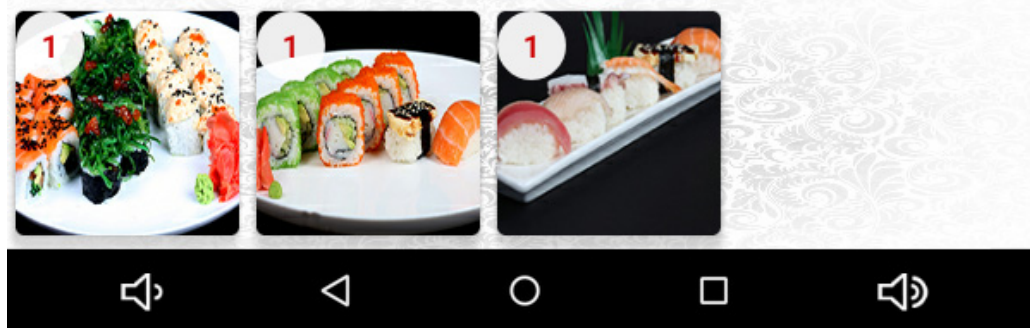


Joonis 18. Tellimuse vaade

3.2.6 Tellimuse eelvaade

Tellimuse eelvaate fragmendi jaoks valis autor *ListView* asemel Androidi Lollipop versioonis tutvustatud uue vidina *RecyclerView*, mis on *ListView* edasiarendatud ja paindlikum versioon. *RecyclerView* sunnib kasutama *ViewHolder* mustrit, mis sisaldab lihtsat klassi, kus hoitakse viiteid *ListView* iga rea kasutajaliidese komponentidele. Sellise mustri abil välditakse kasutajaliidese komponentide otsimist iga kord kui süsteem kuvab loendi rida, tänu millele tõuseb oluliselt loendi kerimise kiirus, eriti just dünaamiliselt muutuvate elementidega loendite puhul (Jim, 2014). *RecyclerView* pakub samuti mõnda atraktiivseid visuaalseid lisasid, näiteks animatsioonid. Peale *RecyclerView* nõude kasutada *ViewHolder* mustrit, on *RecyclerView* rakendamine väga sarnane *ListView* rakendamisega, mida kirjeldati eelpool.

Tellimuse eelvaate fragment kuvab toitude pildid ja toitude kogused, mis saadakse eelpool kirjeldatud `Order` klassi objektist, kus hoitakse tellimuse andmeid (Joonis 19).



Joonis 19. Tellimuse eelvaade

3.3. Keele valiku lahendus

Android pakub mitmeid võimalusi rakenduse andmete salvestamiseks. Kasutaja keele eelistuse salvestamiseks ja saamiseks otsustati kasutada *SharedPreferences* (jagatud eelistused), mis võimaldab andmeid salvestada ja välja otsida võti-väärtus kujul ühe rakenduse ulatuses. Rakendusepõhiste eelistuste saamiseks kasutatakse *Activity* sees *getSharedPreferences()* meetodit. *SharedPreferences* kasutamise eelis seisneb selles, et andmed on püsivad (ei sõltu rakenduse elutsüklist) ja neid saab välja otsida ka siis kui rakendus on vahepeal suletud ja taas avatud. Teiseks eeliseks on ka see, et *SharedPreferences*-ile saab lisada liidese *OnSharedPreferencesChangeListener*, mis lubab reageerida mingi programmeerija poolt määratud tegevusega kui toimub muutus *SharedPreferences* klassis. Antud projektis arendatavas toidu menüü rakenduses on selliseks tegevuseks määratud vastava keele andmete laadimine ja akna vaate värskendamine kui toimub keele vahetus.

3.4. Kohanduv teksti suurus

Arenduse käigus puututi kokku huvitava probleemiga – Androidi raamistikul puudub tugi, et luua *TextView* elementi, mis automaatselt mahutaks teksti vastavalt elemendi suurusele. Mitmetes selles töös arendatava rakenduse elementides, näiteks toitu esindavas loendi reas, ei pruugi toidu pikem kirjeldus *TextView* vidinasse ära mahtuda. Autoril ei õnnestunud leida Android SDK ametlikest materjalidest tuge selle probleemi lahendamiseks. See probleem on laialt teadaolev ja selle üle on palju

arutletud programmeerijate foorumites⁷. Õnneks on olemas mitmeid kolmandate osapoolte teeke, mis aitavad seda probleemi lahendada.

Selles projektis kasutamiseks valiti *AutoFitTextView*⁸ teek, mis võtab tekstielemendi suuruse ja kontrollib, kas näidatud suurusega tekst mahub välja mõõtmetsesse, ning kui ei mahu, siis vähendab fondi suurust. Seda protsessi korratakse, kuni tekst on mahutatud tekstivälja mõõtmetsesse (AndroidDeveloperLB, 2016). Sellisel lähenemisel on ka mõningad puudused. Kui tekst on liiga pikk võib välja mõõtmetsesse mahutatud teksti fondi suurus olla väga väike, kuid selline probleem on osaliselt lahendatav määrates minimaalse lubatud fondi suuruse, kuid sellega on oht kaotada osa teksti kui see ei mahu välja mõõtmetsesse. Veel üks probleem seisneb selle, et kui fondi vähendamise tsüklit peab mitu korda kordama, võib teksti vähendamise protsess olla kasutajale märgatav – see näib justkui teksti vähendamise animatsioonina. Vaatamata mõnede eeltoodud puudustele on selline lahendus siiski aktsepteeritav, kuna see lahendab automaatse tekstisuuruse probleemi.

Kasutamaks teksti suuruse automaatse muutmise võimekust tuleb *AutoFitTextView* teek projekti importida ja standardse *TextView* vidina asemel tuleb kasutada selle teegi *AutoFitTextView* vidinat.

3.5. Rakenduse kasutamine kiosk režiimis

Kiosk režiim või rakenduse pinnimine (ingl *pinning*) ekraanile tähendab, et seade on lukustatud kasutama ainult määratud rakendust ja kasutaja ei saa rakendusest väljuda ega kasutada seadet muudeks tegevusteks (Klein, 2014). Sellise režiimi võimaldamiseks on olemas tasulisi rakendusi, mis pakuvad ka palju erinevaid lisavõimalusi, kuid siin töös arendatava rakenduse jaoks valiti tasuta kiosk režiim, mis on kättesaadav alates Android 5.0 Lollipop versioonist.

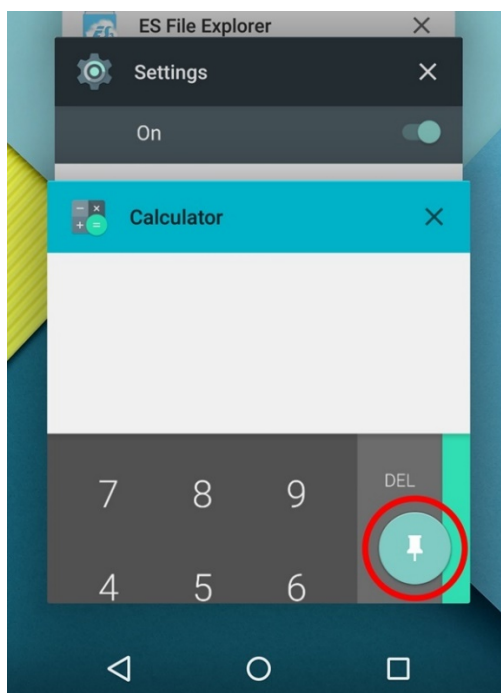
Ekraani pinnimine on sisse lülitatud seadme turvaseadistustes. Seadistuste leidmiseks tuleb avada seadme seadete menüü ja sealt leida alaosa “Turvalisus”, kus tuleb kerida alla täpsemate valikuteni ja vajutada “ekraani pinnimine”, peale mida on pinnimise režiim aktiveeritud.

⁷ Stack Overflow: <http://stackoverflow.com>

⁸ AutoFitTextView: <https://github.com/AndroidDeveloperLB/AutoFitTextView>

Rakenduse pinnimiseks ekraanile, tuleb rakendus esialgu käivitada ja seejärel vajutada “Ülevaate” nupule (ruudukujuline nupp ekraani alumises servas). Ülevaate ekraanil on rakenduse eelvaate nurgal kujutatud roheline nööpnõel (Joonis 20). Kiosk režiimi aktiveerimiseks tuleb vajutada soovitud rakenduse nööpnõela kujutisel. Kiosk režiimist lahkumiseks tuleb korraga vajutada nuppudel “tagasi” ja “ülevaade”.

Turvalisuse huvides on otstarbekas aktiveerida PIN koodiga kaitstud ekraanilukk, misjärel saab valida seadistuse “küsi pinnimise eemaldamiseks koodi”. Sellise seadistusega küsitakse PIN koodi, kui keegi püüab kiosk režiimist lahkuda.



Joonis 20. Rakenduse pinnimine Android seadmes

4. Testimine ja edasiarendus

Arendusfaasi iteratsioonide käigus toimus vigade otsimine ja funktsionaalsuse kontrollimine autori poolt. Arendusfaasi lõpus, kui leitud vead olid parandatud ja rakenduse funktsionaalsus vastas nõuetele, toimus järjekordne kohtumine kliendiga. Kliendil tekkisid rakenduse kujunduse osas mõned täiendavad soovid, mis said samuti rakendatud. Näiteks muudeti esialgne pruun taust halliks ja kujundite ümmargune kujundus asendati kandilisega. Kui klient kinnitas heaks rakenduse disaini ja funktsionaalsuse, viidi läbi rakenduse testimine restorani küllastajate poolt.

4.1. Vigade otsimine

Androidi rakenduste automaatne testimine on keeruline. Androidi arendaja ametliku juhise poolt on soovitatud kasutajaliidese testimiseks kasutada lihtsalt inimest, kes teeks rakendusel läbi rea erinevaid operatsioone, ning kinnitab, et rakenduse käitub vastavalt ettenähtule (Google Inc, kuupäev puudub). Siin töös arendatava rakenduse testimisega tegeles arenduse käigus järjepidevalt autor ning testimisel leitud vead ja puudused parandati kohe.

Arendamise protsessi käigus ilmnis palju vigu, millest olulisemadena võib välja tuua järgnevad: teatud tingimustes kuvas tellimuse eelvaade väärade toote numbrit; pikk tekst ei mahtunud ettenähtud tekstiväljale; vigane *activity* elutsükli haldamine; toodete nimekirja ridadel polnud võimalik vajutada kui rida sisaldas pilti. Kõik mainitud vead said parandatud.

4.2. Nõuetele vastavus

Projekti alguses olid paika pandud nõuded arendatavale rakendusele. Need nõuded on kirjeldatud peatükis 2.2. Järgnevalt on toodud kõik nõuded koos nende staatusega ja kommentaaridega (Tabel 1).

Tabel 1. Nõuete täitmise staatused

Nõue	Staatuse
Rakendus peab olema optimeeritud kasutamiseks GoClever Quantum 2 tahvelarvutitel, mille klient oli eelnevalt hankinud.	Tehtud. Rakenduse arendamiseks ja testimiseks kasutati kliendi tahvelarvutit GoClever Quantum 2.

Nõue	Staatus
Rakendus peab valmima kahes keeles: eesti ja vene.	Tehtud. Rakendus töötab eesti ja vene keeles. Kui venekeelne info puudub, kasutatakse eestikeelseid andmeid.
Toidud peavad olema organiseeritud kategooriatesse.	Tehtud.
Kasutaja saab valida toidu kategooriat.	Tehtud. Avaekraanil saab valida toidu kategooria.
Iga toode peab sisaldama fotot, kirjeldust ja hinda.	Tehtud.
Kasutajal peab olema võimalus valikuid salvestada.	Tehtud. Rakenduses on olemas tellimuse detailne vaade ja tellimuse ülevaade.
Kasutajal peab olema võimalus menüüs tehtud valikuid muuta.	Tehtud. Tooteid saab lisada, eemaldada ja muuta nende koguseid.
Peab olema valik kogu tellimuse kiireks kustutamiseks.	Tehtud. Ekraanil on eraldi nupp kogu tellimuse kustutamiseks.
Rakendus peab toimima ilma internetita ja Wi-Fi ühenduseta. Rakenduse kõik andmed peavad olema salvestatud tahvelarvutisse lokaalselt.	Tehtud. Andmed salvestatakse tahvelarvuti sisemälusse.
Menüü peab olema kergelt hallatav ja uuendatav.	Tehtud. Menüü andmed valmistatakse ette PC abil ja kopeeritakse tahvelarvutisse USB kaudu või kasutades välist mälukaarti.
Kasutajaliidese disainile ja värvilahendusele pole kindlaid nõudmisi, kuid kasutajaliides peab toimima sujuvalt.	Tehtud. Testimise käigus ei tuvastatud probleeme kasutajaliidese kiirusega.
Heliefektid pole vajalikud.	Korras. Rakendus ei kasuta helisid.
Rakendust saab käivitada kiosk-režiimis	Tehtud. Kasutatakse Androidi-sisest funktsionaalsust.

Kõik nõuded said arenduse käigus täidetud. Kõige keerulisem oli autori arvates rakendada nõuet, et kasutajal peab olema võimalus toidu valikuid salvestada ja muuta.

4.3. Testimine klientidega

Rakenduse testimine reaalse kasutajatega on arendusprotsessi üks olulisemaid osi, sest see aitab tagada, et rakendus täidab oma eesmärgi, ja aitab leida vigu, mida programmeerija ise ei pruugi märgata (Setter, 2016).

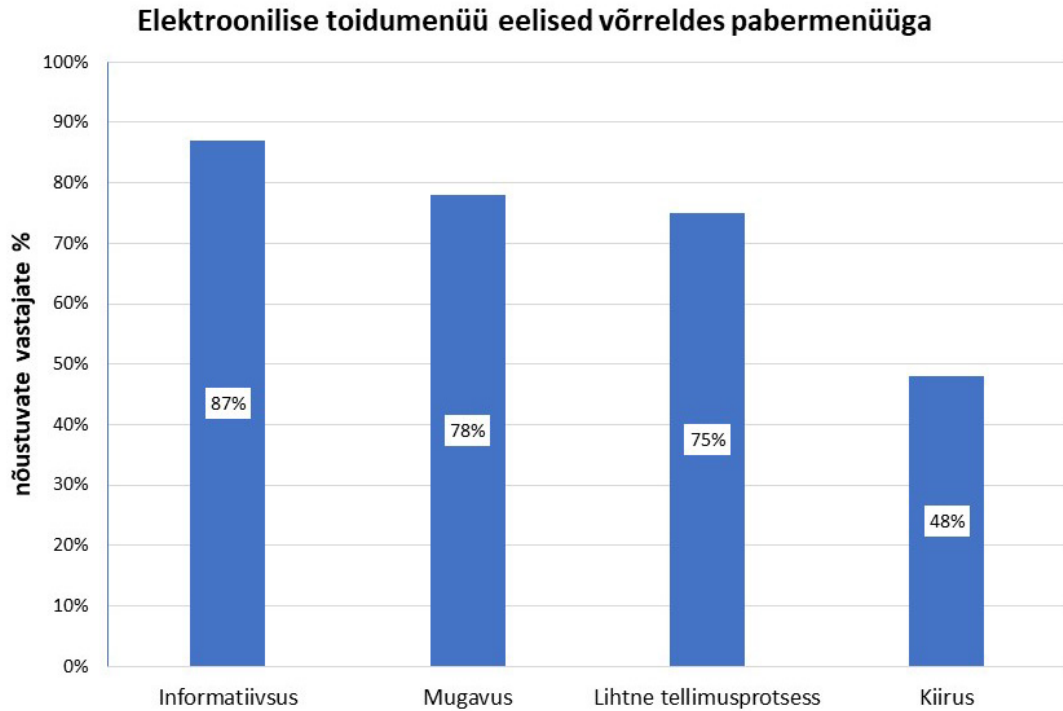
Testimine klientidega teostati tellija restoranis. Testimiseks kulus kaks tundi ja selles osalesid restoranist 30 klienti. Külastajad, kes olid nõus testimises osalema, said

tänuks vastu 10-protsendilise allahindluse enda tellimusest. Iga testimises osalenud külastaja sai restoranis kasutamiseks tahvelarvuti ja lühikese küsimustiku. Küsimustikus oli neli küsimust jah/ei vastusevariantidega ja üks küsimus, kus kasutajatel paluti vabas vormis kirjeldada, milliseid probleeme neil menüü kasutamisel tekkis. Küsimustiku eesmärgiks oli võrrelda elektroonilise menüü omadusi pabermenüü omadustega ja saada vastukaja elektroonilise menüü kasutamise kohta.

Klientidele esitatud küsimused olid järgmised:

1. Kas olete nõus, et elektrooniline menüü on võrreldes pabermenüüga mugavam?
2. Kas olete nõus, et elektrooniline menüü on võrreldes pabermenüüga kiirem?
3. Kas olete nõus, et elektrooniline menüü on võrreldes pabermenüüga informatiivsem?
4. Kas olete nõus, et elektrooniline menüü on võrreldes pabermenüüga lihtsam?
5. Kui Teil esines menüü kasutamisel mingeid probleeme, palun kirjeldage neid.

Küsimustiku tulemused on toodud graafikuna (Joonis 21), mis näitab nõustuvate vastuste osakaalu.



Joonis 21. Elektronilise toidumenüü eelised võrreldes pabermenüüga

Kõige rohkem (87%) rõhutasid vastajad elektronilise menüü informatiivsust. Kõrged hinded said ka mugavus (78%) ja tellimusprotsessi lihtsus (75%). Saadud tulemused on sarnased varasemalt töös välja toodud statistikaga (ptk 1.2), kus uuriti tahvelarvutite kasutamise eeliseid USA restoranides. Samas ainult 48% vastajatest arvasid, et elektronilise menüü kasutamine on pabermenüüga võrreldes kiirem. Selline vastuste jaotus võib olla tingitud sellest, et elektroniliste menüüde kasutuselevõtmine võib esialgu põhjustada teeninduse aeglustumist, kuna klientidel on vaja aega nendega harjumiseks (Hill, 2015).

Tuginedes kasutajatelt saadud vastukajale võib väita, et rakendus võeti positiivselt vastu ja see täitis tellija nõudmised. Siiski, kasutajatega testimisel ilmnisid ka mõned puudused:

- Prügikasti ikooniga kujutatud tellimuse kustutamise nupp ja silma ikooniga kujutatud tellimuse eelvaate nupp põhjustasid mõningat segadust, kuna mõned kliendid ei mõistnud esialgu, mida need nupud tähendavad.
- Kasutajad eeldasid, et tellimuse eelvaate sektsioon on vajutatav ja see võiks avada tellimuse lehe, kuid tegelikult pole see alumine tellimuse eelvaate

seksioon vajutatav – tellimuse vaatamiseks tuleb vajutada silma ikooniga nupule.

Vaatamata mõnede leitud probleemidele, oli tellija valminud rakendusega rahul ja rakendus anti üle tellijale. Otsustati, et reaalelu testimise käigus esile tulnud puudused ei takista rakenduse kasutamist, ning neid puuduseid on võimalik parandada järgmise edasiarenduse tsükli ajal.

4.4. Edasiarendamise ideed

Järgmiselt on kirjeldatud mõned mõtted, mis tekkisid autoril antud bakalaureusetöös arendatud tarkvara edasiarendamiseks.

1. Prügikasti ja silma ikoonidega nupud oleks otstarbekas muuta tekstiga nuppudeks, et parandada nende nuppude tähenduste arusaadavust ja vältida võimalikke segadusi.
2. Alumise tellimuse eelvaate seksiooni vajutamine võiks avada tellimuse ülevaate akna, nagu kasutajad intuiivselt eeldasid.
3. Kõik *ViewList* vidinad asendada uuemate *RecyclerView* vidinatega, mis on ressursside kasutamisel efektiivsemad. Seeläbi paraneks seadme aku ajaline kestvus.
4. Lisada keskse haldussüsteemi võimalus võrgu kaudu, mis võimaldaks teha muudatusi toidumenüü andmetes kõikides seadmetes korraga.
5. Lisada võimalus tellimuse saatmiseks teenindusterminali.

Punktid, mis on tähistatud nimekirjas numbritega 1 ja 2, on kõrgeima prioriteediga ja nende rakendamine ei nõua põhilist rakenduse programmeerimiskoodi muutmist. Võrguühendus ja keskne haldussüsteem ei ole rakenduse tellijale kriitilised ja samas nõuaksid täiendavat uurimist ja töömahukat tarkvara programmeerimise koodi täiendamist. Autori arvates oleks tellijal mõistlik kasutuses olevaid tahvelarvutid välja vahetada parema ekraaniga ja jõudlusega mudeli vastu. Rakenduse tellijale sai tehtud selline ettepanek, millega ta ka nõustus.

Kokkuvõte

Käesolev bakalaureusetöö lähtus probleemist, et toitlustusettevõtetes kasutatavad pabermenüüd on tehnoloogia arenguga mõnevõrra iganenud ja nende uuendamine on keeruline, samas eeldavad tänapäevased külastajad ajakohasust ja informatiivsust ka toidumenüüdest. Käesoleva bakalaureusetöö eesmärgiks oli arendada tarkvara, mis võimaldaks asendada toitlustusettevõtetes pabermenüüd kaasaegsete elektrooniliste vastu kasutades tahvelarvuteid, ning kirjeldada selle tarkvara arendusprotsessi.

Eesmärgi saavutamiseks määratleti kõigepealt antud töö raames arendatava projekti struktuur – mida ja millises järjekorras teha. Töös püstitatud probleemist paremaks arusaamiseks tehti analüüs tahvelarvutite kasutamisest restoranides. Uuriti sarnaseid olemasolevaid rakendusi, et selgitada välja, milliseid omadusi võiks projektis kasutada ja milliseid vältida. Lisaks anti ülevaade toidumenüü arendamise ja disainimise juhistest ning Androidi ametlikest juhistest.

Rakenduse disainimise etapis selgiti välja, millistele nõuetele peab valmiv rakendus vastama ning vastavalt sellele otsustati arendatava tarkvara funktsionaalsus. Kasutades sisendiks varasemalt kirjeldatud funktsionaalsuse nõudmisi, menüü kujundamise juhendeid ning teisi sarnaste rakenduste häid omadusi, loodi rakenduse kasutajaliidesega vaadete visandid.

Arenduse etapi käigus programmeeriti digitaalse menüü tarkvara lähtudes disainimise etapis valminud visanditest ja toetudes varem vaadeldud juhistele. Samas kirjeldati arenduse käigus tekkinud põhilisi probleeme ja nende lahendusi.

Valminud toidumenüü rakendust testiti ja hinnati mitmes etapis. Esmalt testiti tarkvara autori poolt arenduse etapi jooksul ja kui rakenduse funktsionaalsus oli realiseeritud, viidi läbi tarkvara testimine restorani küllastajate poolt. Parema tagasiside saamiseks paluti testimises osalejatel vastata lühikesele küsimustikule. Selle eesmärgiks oli uurida, kas valmis rakendus pakub paremaid võimalusi võrreldes tavaliste pabermenüüdega ning seega täidab oma eesmärgi. Tagasiside saamine kasutajatelt oli väga tähtis, kuna seeläbi sai tuvastada mõned rakenduse puudused, mis jäid autorile endale märkamata ning mida võiks tulevikus parandada.

Käesoleva töö raames sai valmis elektrooniline toidumenüü, mida saab kasutada Android platvormiga tahvelarvutitel. Lisaks kirjeldati kõiki rakenduse arendamise protsessi etappe ja anti ülevaade selle käigus tekkinud põhilistest probleemidest ning nende lahendustest. Restorani klientidega testimise järelendusena võib autor väita, et valmis rakendus vastab püstitatud eesmärkidele.

Bakalaurusetöö andis autorile palju uusi teadmisi kasutajaliidese kavandamise ja Android platvormile arendamise osas, suurendades sellega edasist huvi selle valdkonna vastu.

Kasutatud kirjandus

- AndroidDeveloperLB. (12. juuli 2016. a.). *AutoFitTextView Readme*. Allikas: GitHub:
<https://github.com/AndroidDeveloperLB/AutoFitTextView>
- Creger, R. (2014). *8 essential restaurant menu design tips*. Allikas: 99designs:
<https://99designs.com/blog/tips/menu-design-roundup-tips/>
- Gentile, D. (07. september 2014. a.). *The 11 untold secrets of menu design*. Allikas: Thrillist: <https://www.thrillist.com/eat/nation/restaurant-menu-secrets-menu-design>
- Google Inc. (kuupäev puudub). *Android Studio User Guide*. Allikas: Android Studio User Guide: <https://developer.android.com/studio/intro/index.html>
- Hill, S. (14. mai 2015. a.). *Restaurant table tablets: A gimmick, or actually helpful?* Allikas: The Christian Science Monitor: <http://www.csmonitor.com/Business/Saving-Money/2015/0514/Restaurant-table-tablets-A-gimmick-or-actually-helpful>
- Jim, W. (14. detsember 2014. a.). *Android (v5) Lollipop RecyclerView Tutorial*. Allikas: Intertech: <https://www.intertech.com/Blog/android-v5-lollipop-recyclerview-tutorial/>
- Klein, M. (22. november 2014. a.). *How to Pin Screens in Android 5.0 for More Security and Privacy*. Allikas: How-to Geeks: <https://www.howtogeek.com/202841/how-to-pin-screens-in-android-5.0-for-more-security-and-privacy/>
- Lars, V. (9. november 2016. a.). *Using lists in Android with ListView - Tutorial*. Allikas: Vogella: <http://www.vogella.com/tutorials/AndroidListView/article.html>
- Mehul, R. (20. mai 2015. a.). *Why Android Studio Is Better For Android Developers Instead Of Eclipse*. Allikas: DZone: <https://dzone.com/articles/why-android-studio-better>

- Pavesic, D. (kuupäev puudub). *The Psychology of Menu Design: Reinvent Your 'Silent Salesperson' to Increase Check Averages and Guest Loyalty*. Allikas: Restaurant Resource Group: http://rrgconsulting.com/psychology_of_restaurant_menu_design.htm
- Setter, M. (2016). *User Acceptance Testing – How To Do It Right!* Allikas: Usersnap: <https://usersnap.com/blog/user-acceptance-testing-right/>
- Statista. (märts 2014. a.). *Attributes of tabletop tablets in United States restaurants as of March 2014*. Allikas: Statista: <https://www.statista.com/statistics/301109/attributes-of-tabletop-tablets-in-restaurants-us/>
- Williams, R., & Julian, J. (22. jaanuar 2015. a.). *5 Reasons Your Restaurant Needs a Table Ordering System*. Allikas: The Restaurant Technology Guys: <http://restauranttechnologyguys.com/5-reasons-restaurant-needs-table-ordering-system/>

Summary

Development of a Restaurant Menu Application on Android Platform

Bachelor's Thesis

The goal of this Bachelor's thesis was to develop a restaurant menu application on Android platform and to describe the development process.

To reach the goal, the following questions were researched and answered:

- what are the general principles of designing a food menu
- what technological means exist to implement the desired functionality on Android platform
- what are the strengths and weaknesses of the developed application identified during its testing

The first chapter of this thesis describes the project preparation phase, during which the problem background was researched. In the second chapter, the application design process is described. During that process, the application visual prototypes were created. The third chapter is dedicated to the application development cycle, during which the application functionality was programmed. The chapter gives an overview of some problems encountered during the cycle. The fourth and closing chapter describes the testing process of the created restaurant menu application. Based on the testing results some ideas for application future development are introduced.