

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Informaatikainstituut
Tarkvaratehnika õppetool

Müügiandmete visualiseerimine ettevõtte aruandlusmooduli näitel

Bakalaureusetöö

Üliõpilane:	Kristel Palts
Üliõpilaskood:	121115IABB
Juhendaja:	Tarvo Treier

Tallinn
2015

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

(kuupäev)

(allkiri)

Annotatsioon

Andmete visualiseerimine mängib ettevõttes olulist rolli andmete mõistmise ja analüüsi juures, kuid tänapäeval on visualiseerimisteede valik üsna kirju ja see teeb konkreetse ülesande jaoks sobivaima teegi valimise keeruliseks.

Käesolevas töös on uuritud erinevaid *JavaScript*'i andmete visualiseerimise teeke, võrreldud nende võimalusi ja selle tulemusena valitud välja sobivaim teek ettevõtte aruandlusmooduli realiseerimiseks.

Sobivaima *JavaScript*'i teegi valimiseks on kokku kogutud ettevõtte aruandlusmooduli nõuded, neist tuletatud erinevad omadused ja funktsionaalsused, mida teegid peaksid täitma, ja nende põhjal võrreldud teekide võimalusi. Võrdluse tulemusel leitud kahe sobivaima teegiga on realiseeritud aruandlusmooduli diagrammide prototüübid, neid analüüsitud ja nende põhjal lõppjärelused tehtud.

Lõputöö on kirjutatud eesti keeles, ning sisaldab teksti 42 leheküljel, 3 peatükki, 10 joonist, 6 tabelit.

Abstract

Data visualization plays an important role in understanding and analyzing company's data. Nowadays there is a great choice of different data visualization libraries, which makes it difficult to choose the most suitable library for a specific task.

Current thesis explores different *JavaScript* data visualization libraries, compares their possibilities and as a result chooses the most suitable library for creating a company's reporting module.

For selecting the most suitable *JavaScript* library, the requirements for company's reporting module are collected, different features and functionalities are derived from them, which libraries should perform, and finally, the libraries are compared on the basis of these features and functionalities. Two of the most suitable libraries, which turned out from the previous comparison, are used to create the prototypes of required reporting module's graphs. The final decision is made on the basis of the analysis of prototypes.

The thesis is written in Estonian and contains 42 pages of text, 3 chapters, 10 figures and 6 tables.

Lühendite ja mõistete sõnastik

Andmete visualiseerimine - *Data visualization*. Andmete visualiseerimine ehk andmete visuaalne kujutamine on andmete esitlemine pildi- või graafilises formaadis, mille eesmärgiks on pakkuda kasutajale paremat arusaama andmete sisust.

SVG – *Scalable Vector Graphics*. Skaleeritav vektorgraafika – keel 2-dimensiooniliste graafikute kirjeldamiseks.

XML – *Extensible Markup Language*. Laiendatav märgistuskeel veebidokumentide kirjutamiseks.

HTML – *HyperText Markup Language*. Standardne märgistuskeel veebilehtede loomiseks.

DOM – *Document Object Model*. Rakenduste programmeerimise liides HTML ja XML dokumentide jaoks, mis võimaldab dünaamiliselt uuendada dokumendi sisu, struktuuri ja stiili.

CSS – *Cascading Style Sheets*. Astmelised stiililehed märgistuskeeles kirjutatud dokumentide välimuse ja vormingu kirjeldamiseks.

D3 – *Data Driven Documents*. *JavaScript*'i visualiseerimisteek, mis kujutab endast uudset lähenemist dünaamiliste ja interaktiivsete visualisatsioonide kuvamiseks veebis.

Jooniste nimekiri

Joonis 1 Andmete visualiseerimise põhisammud.....	12
Joonis 2 Ettevõtte visand aruandlusmoodulist	23
Joonis 3 Pakitud ja rühmitatud tulpadega diagramm.....	25
Joonis 4 Võrdlustabeli tulemuste võrdlus.....	30
Joonis 5 Tulpdiagramm kuude müügist (Google Charts).....	34
Joonis 6 Tulpdiagramm kuude müügist (D3)	34
Joonis 7 Joondiagramm viimase 15 päeva müügist (Google Charts).....	35
Joonis 8 Joondiagramm viimase 15 päeva müügist (D3)	35
Joonis 9 Tulpdiagramm aastate müügist (Google Charts).....	36
Joonis 10 Tulpdiagramm aastate müügist (D3).....	36

Tabelite nimekiri

Tabel 1 Canvase ja SVG võrdlus.....	14
Tabel 2 Teekide võrdlus	27
Tabel 3 Võrdlustabeli tulemused protsentidena	29
Tabel 4 Tähelepanekud kuude müügi tulpdiagrammide prototüüpidest	34
Tabel 5 Tähelepanekud viimase 15 päeva müügi joondiagrammi prototüübist.....	35
Tabel 6 Tähelepanekud aastate müügi tulpdiagrammi prototüübist.....	37

Sisukord

Sissejuhatus	9
1. Põhiteadmiste ülevaade	11
1.1 Andmete visualiseerimine.....	11
1.2 Visualiseerimise teegid	17
1.2.1 D3 – Data-Driven Documents	17
1.2.2 Elycharts	18
1.2.3 Google Charts	18
1.2.4 SVGware	20
2. Nõuded aruandlusmoodulile ja sobivaima teegi valimine.....	21
2.1 Senised lahendused	21
2.2 Metoodika sobivaima teegi leidmiseks	22
2.3 Aruandlusmooduli nõuded.....	22
2.4 Teekide võrdlus.....	27
3. Aruandlusmooduli osade prototüübid	33
3.1 Prototüübid ja nende analüüs	33
3.2 Järeldused.....	37
Kokkuvõte	39
Kasutatud kirjandus	41

Sissejuhatus

Andmete visualiseerimine mängib ettevõttes olulist rolli andmete mõistmise ja analüüsi juures. Käesoleva töö autorile anti ettevõtte poolt ülesandeks luua nende müügiandmeid visualiseeriv aruandlusmoodul, kasutades selleks *JavaScript*'i visualiseerimisteeki. *JavaScript*'i visualiseerimisteekide valik on aga üsna lai, seetõttu on oluline leida neist sobivaim antud ülesande teostamiseks.

Käesolev töö kujutab endast sobivaima *JavaScript*'i visualiseerimisteegi leidmist ettevõtte aruandlusmooduli realiseerimiseks.

Kavandatava aruandlusmooduli kasutajaks on liisinguettevõtte, mille põhitegevuseks on pakkuda inimestele võimalust soetada erinevaid kaupasid järelmaksuga. Liisingut pakutakse jaepoodides, internetipoodides ja kataloogikaupade tellimiskeskustes. Liisingulepinguid vormistavad kaupluste müüjad. Hiljem tasub ostja kauba eest osamaksete kaupa liisinguettevõttele. Hetkel on koostööpartnereid (poegruppe) enam kui 500 ning müügikohti enam kui 1500 üle Eesti. Klientide arv ületab 200 000.

Liisinguettevõtte jaoks on väga oluline olla kursis müüginumbrite ja –tulemustega. Selle baasil hinnatakse tulemuslikkust ning rahavoogusid. Ettevõtte töötajatel tekkis vajadus ühtse visuaalse ülevaate järele kõige tähtsamatest müügiga seotud numbritest. Seni on kogu informatsioon saadud kätte mahukates tabelites olevatest numbritest, mis ei anna korraga piisavalt efektiivset, kiiret ja laialdast ülevaadet ettevõttega seotud müüginumbritest. Müüginumbrite visuaalne kujutamine aitab tunduvalt paremini mõista nende proportsioone, mustreid ja seoseid. Andmetega pidevalt kursis olemine ja nendest ühtlasi ka aru saamine aitab töötajatel paremini ja õiges suunas oma töösse panustada ning juhtidel veenduda oma äri edukuses või leida erinevaid probleemseid kohti, mida parandada. Kõik see aitab suures määras kaasa äri jätkusuutlikkusele.

Käesoleva töö autori ülesandeks on leida sobivaim *JavaScript*'i visualiseerimisteek aruandlusmooduli realiseerimiseks. Moodul peab vastama ettevõtte ootustele ja seatud nõuetele, olema seotud juba olemasoleva lepingute haldamise süsteemiga ning arvestama hilisema täiendamise ning laiendamise võimalusega.

Eesmärk

Lõputöö eesmärgiks on leida ettevõtte aruandlusmooduli realiseerimiseks sobivaim *JavaScript*'i andmete visualiseerimise teek.

Ülesanded

Töö eesmärgini jõudmiseks tuleb autoril esmalt tutvuda andmete visualiseerimise teemaga üldiselt ning anda ülevaade *JavaScript*'i visualiseerimisteedidest, mille hulgast ta hakkab aruandlusmooduli realiseerimiseks sobivaimat valima.

Teiseks tuleb anda ülevaade aruandlusmoodulist ja selle nõuetest, tuletada neist omadused ja funktsionaalsused, mille põhjal erinevate *JavaScript*'i teekide võimalusi võrrelda ning esitada nende põhjal teekide võrdlustabel. Võrdlustabeli tulemusena filtreerib töö autor välja 2 kõige sobivamat teeki.

Kolmandaks tuleb kahe sobivaima *JavaScript*'i teegiga luua aruandlusmooduli nõuetele võimalikult ligilähedased diagrammide prototüübid, neid analüüsida ja teha järeldused.

Ülevaade tööst

Esimeses peatükis tutvustab töö autor andmete visualiseerimist üldiselt, et saada ülevaadet olulistest aspektidest, sammudest ja mõistetest andmete visualiseerimise teema juures. Samuti teeb autor valikud tehnoloogiate osas, mis võetakse aluseks *JavaScript*'i visualiseerimisteedide valimisel edasise analüüsi jaoks. Seejärel annab autor ülevaate valituks osutunud neljast *JavaScript*'i visualiseerimisteedist, et tutvuda nende erinevate võimaluste ja eripäradega.

Teises peatükis annab töö autor täpsema ülevaate realiseeritavast aruandlusmoodulist ja sellele esitatud nõuetest. Seejärel võrdleb autor eelnevalt käsitletud *JavaScript*'i teekide võimalusi vastavalt aruandlusmoodulile esitatud nõuetele ning valib välja kaks sobivaimat teeki.

Kolmandas osas esitab autor kahe sobivaima *JavaScript*'i teegi abil loodud aruandlusmooduli diagrammide prototüübid, analüüsib neid ja teeb järeldused.

1. Põhiteadmiste ülevaade

Käesolevas peatükis tutvustatakse andmete visualiseerimist üldiselt, et saada ülevaadet olulistest aspektidest, sammudest ja mõistetest andmete visualiseerimise teema juures. Samuti tehakse valikud tehnoloogiate osas, mis võetakse aluseks *JavaScript*'i visualiseerimisteede valimisel edasise analüüsi jaoks. Seejärel antakse ülevaade valituks osutunud neljast *JavaScript*'i visualiseerimisteedest, et tutvuda nende erinevate võimaluste ja eripäradega.

1.1 Andmete visualiseerimine

Andmete visuaalne kujutamine on andmete esitlemine pildi- või graafilises formaadis ning selle eesmärgiks on pakkuda kasutajale paremat arusaama andmete sisust.

Andmete visualiseerimine mängib olulist rolli andmete analüüsi juures. Eelkõige lihtsustab see suurte andmemahtude mõistmist, kuna läbi graafikute on andmete proportsioonid inimsilmale vastuvõetavamad ja neid on mugavam jälgida. Samuti võib graafikutelt kergemini välja lugeda andmehulga olulisi mustreid ja seoseid, mille põhjal teha erinevaid äriotsuseid ja valikuid ning leida ka probleemseid kohti. Lisaks sellele tagab informatsiooni saamine visuaalsel kujul ka olulise aja kokkuhoiu, võrreldes teksti või numbrite läbivaatamisega, sest graafikud rõhutavad andmebaasis olevat olulist informatsiooni tunduvalt arusaadavamal kujul kui tekstiline esitus.

Kokkuvõtlikult võibki välja tuua 4 olulist andmete visualiseerimise hüve [1]:

- Võimaldab kergesti analüüsida trende/mustreid
- Võimaldab leida kriitilist informatsiooni
- Võimaldab mõista seoseid andmete vahel
- Võimaldab andmetega suhelda

Need tunnused muudavadki andmete visualiseerimise tänapäeva andmerakendustes üha enam vajaduseks, mitte enam valikuks [1].

Andmete visualiseerimine nõuab oskusi nii kujunduse, veebiarenduse, programmeerimise kui ka andmebaasi valdkonnas.

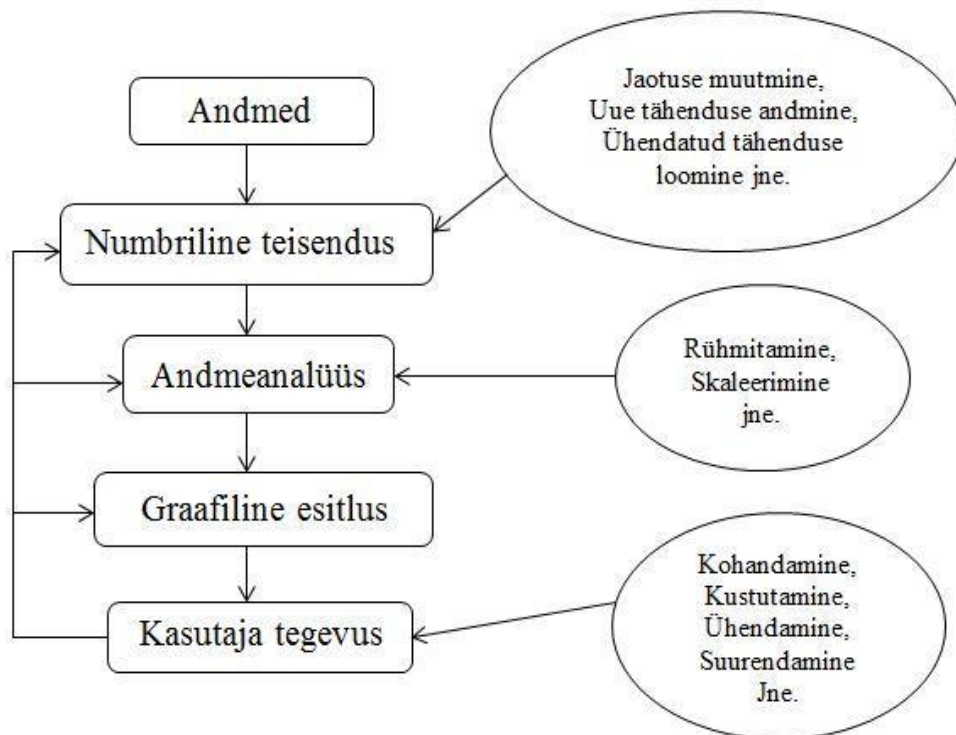
Andmete graafiline esitlemine hõlmab graafilisi osade kasutamist nagu punktid, jooned (kurvid), kujundid, pildid ja tekst, aga ka atribuute nagu värvus/intensiivsus, suurus, asetus ja kuju. [2]

Andmetest aru saamine hõlmab nii tajumist, hindamist kui ka võrdlemist [2].

Kõige olulisemad üldised nõuded andmete visualiseerimise puhul [3]:

- Tuleb aru saada andmetest, mida hakatakse visualiseerima, sealhulgas andmete mahust ja kardinaalsusest.
- Tuleb kindlaks määrata, mida üritatakse visualiseerida, mis tüüpi informatsiooni ja millisel kujul tahetakse edasi anda.
- Tuleb tunda kasutajaid/publikut ehk kellele saavad need visualiseeritud andmed olema suunatud ning mõista, kuidas nad visuaalset informatsiooni tõlgendavad.
- Kasutada visualiseerimisvahendit, mis kõige paremini ja lihtsamini annaks kasutajale infot edasi.

Järgneval skeemil on kujutatud andmete visualiseerimise põhisammud [4]:



Joonis 1 Andmete visualiseerimise põhisammud

- **Numbriline teisendus [4]**

Visualiseerimine on üldjuhul numbriliste andmete teisendamine. Numbrid on abstraktses mõistes ning nende esitlemine punktide, joonte ja tulpadena nõuab nende teisendamist.

Teisendamine hõlmab:

- Jaotuse muutmist: muuta numbrite jaotust nii, et nad sobiksid paremini analüüsiks või visuaalseks esitluseks. Muutmine võib olla nt lineaarne, logaritmiline, normaliseeriv vms.
- Täheenduse ümberdefineerimine: kohandada numbreid nii, et nad tähendaksid midagi või annaksid parema ettekujutuse soovitud sisust.
- Summeeritud täheenduse loomine.

- **Andmeanalüüs [4]**

Andmeanalüüs on erinevate meetodite rakendamine andmetele, et aidata kaasa nende tõlgendamisel. Mõned uurimuslikud andmeanalüüsi meetodid on näiteks statistiline tugi, klasteranalüüs, mitmedimensiooniline ulatus ning faktoranalüüs. Andmeanalüüsi võidakse kasutada, et andmeid teisendada või summeerida andmeid või nende statistilisi omadusi.

- **Graafiline esitus [4]**

Graafiline esitus koosneb mõnest võtmetegevusest nagu suurusjärgu hindamisest, proportsioonide hindamisest, trendide ja kallete hindamisest ning grupeerimise hindamisest.

- **Kasutaja tegevus [4]**

Visualiseerimise tulemuse esitlemisel võivad kasutajad leida, et tulemused ei kattu nende ootustega. Kasutajad võivad soovida mõningaid muutusi, mis nõuavad varasemate sammude kordamist, näiteks kohandada visuaalsete objektide asetust, kustutada segadusttekitavaid kohti, tuua esile rohkem tähelepanu nõudvaid osasid jms.

SVG ja Canvas tehnoloogiad

Graafikute määratlemiseks veebis kasutatakse kahte põhilist tehnoloogiat: SVG ja Canvas.

SVG ehk skaleeritav vektorgraafika on keel 2-dimensiooniliste graafikute kirjeldamiseks XML-is (*Extensible Markup Language*). Kuna SVG toimib XML-i baasil, siis iga element on saadaval SVG DOM-is (*Document Object Model*), mis võimaldab dünaamiliselt uuendada dokumendi sisu, struktuuri ja stiili. Seega saab SVG elementidega siduda *JavaScript*'i sündmusekäsitlejaid ning elementide väljanägemist kujundada CSSis (*Cascading Style Sheets*). Iga loodud kuju jäetakse objektina meelde, seega, kui SVG objekti atribuudid muutuvad, siis saab veebilehitseja automaatselt SVG objekti kuju ära muuta. [5]

HTML5 Canvas ehk `<canvas>` elementi kasutatakse graafikute loomiseks skripti kaudu. Canvas elemendil pole eraldi joonistusvõimeid, see on vaid konteiner graafikute jaoks, seega peab graafikute joonistamiseks kasutama skripti [6]. Canvas luuakse pikselhaaval. Kui graafik on valmis joonistatud, siis see unustatakse veebilehitseja poolt. Seega, kui soovitakse graafiku asukohta muuta, siis peab kogu pildi uuesti joonistama, sealhulgas ka kõik need objektid, mis võisid olla graafiku poolt kaetud. [5]

Järgnevas tabelis on esitatud SVG ja Canvase võrdlus [5][7]:

Tabel 1 Canvase ja SVG võrdlus

Canvas	SVG
Sõltuv resolutsioonist	Sõltumatu resolutsioonist
Ei toeta sündmusekäsitlejaid (<i>event handlers</i>)	Toetab sündmusekäsitlejaid (<i>event handlers</i>) – elementidega saab kergesti siduda animatsioone
Vähesed võimalused teksti visualiseerimiseks	Toetab teksti visualiseerimist
Tulemuspildi saab salvestada .png või .jpg formaadis	Täielik kontroll iga elemendi üle, kasutades SVG DOM API-t <i>JavaScript</i> 'is
Pidev jõudlus – kõik on esitatud pikslitena.	Dokumendi keerukuse suurenedes, muutub kõik aeglasemaks, kuna SVG on seotud DOM-iga.

Canvas	SVG
Sobib interaktiivseks pilditötluseks, rastergraafika loomiseks, pildianalüüsiks, mängugraafikaks.	Sobib veebirakenduste kasutajaliidestest kasutamiseks, sest SVG on XML formaadis ja seega dokumentide kättesaadavus veebilehitsejates on tunduvalt parem kui Canvas elementide puhul
Ei sobi veebilehtedele või rakenduste kasutajaliidestesse, sest Canvas nõuab iga elemendi manuaalset ümberjoonistamist liideses.	Ei sobi mängurakendustes kasutamiseks.

Kuna realiseeritav aruandlusmoodul on osa veebirakendusest, peab see olema kuvatav erinevate resolutsioonidega, nõuab ligipääsetavust erinevatele elementidele eraldi ning nõuab ka teksti kuvamist, siis edasised valikud on tehtud SVG tehnoloogial põhinevate teekide kasuks.

JavaScript

JavaScript on kerge interpreteeritav objektorienteeritud programmeerimiskeel, mida kasutatakse interaktiivsete veebilehtede loomiseks ning erinevate veebilahenduste realiseerimiseks. [8]

JavaScript'i keel jaguneb [9]

- Tuum (*core*) *JavaScript* – sisaldab tuumikobjekte nagu *Array*, *Date* ja *Math* ning tuumik-keelelemente nagu operaatorid, kontrollstruktuurid ja deklaratsoonid. Tuum-*JavaScript*'i saab laiendada erinevatel põhjustel, täiendades seda lisaobjektidega.
- Kliendi-poolne (*client-side*) *JavaScript* – laiendab tuumkeelt, täiendades seda objektidega, mis kontrollivad brauserit ja selle dokumendi objekti mudelit (DOM – *Document Object Model*). Kliendipoolsed laiendused võimaldavad rakendusel paigutada elemente HTML vormile ja reageerida kasutaja sündmustele nagu hiireklõpsud, vormi sisestus ja lehe navigeerimine.

- Serveri-poolne (*server-side*) *JavaScript* – laiendab tuumkeelt, täiendades seda objektidega, mida on vaja *JavaScript*'i jooksutamiseks serveril. Serveripoolsed laiendused võimaldavad rakendusel suhelda relatsioonilise andmebaasiga, teostada failide käsitlemist serveril jms.

JavaScript'i kasutades on võimalik luua dünaamilisi HTML veebilehti, mis töötlevad kasutaja sisendeid ja säilitavad püsivaid andmeid, kasutades spetsiaalseid objekte, faile ja relatsioonilisi andmebaase.

JavaScript'i kasutamise eelised [8]:

- Vähem serveriga suhtlemist: kasutaja sisendit saab valideerida enne selle saatmist serverile. See vähendab serverivahelist liiklust, mis tähendab ka väiksemat koormust serverile.
- Kohene tagasiside kasutajatele: nad ei pea ootama lehe uuesti laadimist, et näha, kas nad unustasid midagi sisestada.
- Suurenenud interaktiivsus: on võimalik luua liideseid, mis reageerivad kasutaja tegevusele.
- Külluslikud liidesed: *JavaScript*'iga saab lisada komponentide liigutamist, liigureid jms.

JavaScript on üks kolmest keelest, mida veebiarendajad peavad teadma [10]:

- 1) HTML – veebilehtede sisu määratlemiseks.
- 2) CSS – veebilehe asetuse määratlemiseks.
- 3) *JavaScript* – veebilehtede käitumise programmeerimiseks.

JavaScript on ühtlasi ka platvormist sõltumatu, mis muudab selle väga laialdaselt kasutatavaks. Seetõttu ei ole ka imeks pandav see, et *JavaScript* kuulub maailma kõige populaarsemate programmeerimiskeelte hulka [11].

Kuna realiseeritav aruandlusmoodul saab olema osa veebirakendusena toimivast süsteemist, kus kasutatakse *JavaScript*'i, ning aruandlusmoodul nõuab ka teatavat interaktiivsust, siis visualiseerimisteede valikud on tehtud just tingimusel, et tegemist oleks *JavaScript*'i teekidega.

1.2 Visualiseerimise teegid

Järgnevas alapeatükis tutvustatakse nelja erinevat *JavaScript*'i andmete visualiseerimise teeki: D3 (Data Driven Documents), GoogleCharts, SVGware ja Elycharts. Antud teegid on valitud erinevate teekide võrdlustabelist [12]. Valiku tegemisel on lähtutud nõuetest, et tegemist oleks *JavaScript*'i vabavaralise teegiga ning graafikutehnoloogiaga SVG. Samuti on lähtutud võrdlustabelis [12] välja toodud näitediagrammidest ja teekide kohta käiva info kättesaadavusest.

1.2.1 D3 – Data-Driven Documents

D3 (Data Driven Documents) on vabavaraline *JavaScript*'i teek, mida kasutatakse tavaliselt SVG- ehk skaleeritavate vektorgraafikute loomiseks. SVG ühendab disaini ja programmeerimise. *JavaScript*'i teegina võimaldab D3 koostööd teha paljude teiste vahenditega, milleks on peamiselt HTML ja CSS. D3 ühildub hästi brauseritega, kuna kaasaegsed brauserid ei pea tegema midagi eraldi selleks, et D3'e kasutada, vaid peavad lihtsalt toetama seda raamistikku, mida Internet on omaks võtnud kasutusmugavuse jaoks. [13]

D3 võimaldab seostada andmeid ja ekraanil ilmuvat nii, et need kaks ühendatakse koheselt. See tähendab, et kui muuta andmeid, muutub ka objekt ekraanil. Samuti võimaldab D3 määrata, mis täpselt ekraanil ilmuma peaks – joon, punkt, ring, graafik jpm. [13]

D3 on loodud selleks, et käsitseda graafilisi objekte veebilehel. Selleks, et käsitseda ja filtreerida andmeid, on võimalik D3'ga siduda Crossfilter teek. Neid kahte mittetriviaalset tehnoloogiat on aga keeruline kombineerida. Selle takistuse ületamiseks, et saada andmed veebilehele, kasutatakse lisaks Dc ehk Dimensional Charting teeki. Dc on graafikute teek, mis kasutab ära Crossfilter'i andmete filtreerimise võimalused ja integreerib need D3 graafiliste võimetega. [13]

Positiivsed küljed [14] [15]:

- Võimaldab luua visuaalseid kujutisi ja graafikuid täpselt vajadustele, olgu need ükskõik kui spetsiifiliste nõuetega.

- Pakub väga palju näiteid ja laiendusi peaaegu igat tüüpi graafikutele, mida soovitakse luua.
- Põhjaliku ja hea dokumentatsiooni olemasolu
- Võimaldab tõrgeteta töötada suuremahuliste andmetega, mis varieeruvad bittitest gigabittideni
- Isegi kõige üldisematele graafikutele saab lisada DOM funktsioone nagu suurendamine, klikkimine jms.

Negatiivsed küljed [14] [15]:

- Keerulisemate graafikute puhul võib andmete sidumine *JavaScript*'is olla veidi keeruline.
- Iga telje, joone ja tulba peab looma nullist.
- Nõuab vähemalt põhiteamisi *JavaScript*'ist, et alustada graafikute loomist. Samuti võib D3.js õppimine võtta oodatust kauem aega.
- D3.js ei toeta otse graafikute konverteerimist .png formaati.

1.2.2 Elycharts

Elycharts on kergesti kasutatav ja kohandatav teek diagrammide loomiseks ainult *JavaScript*'iga. Elycharts teek toetab populaarsemaid diagrammitüüpe nagu joon-, (vertikaalset/horisontaalset) tulp-, kiht- ja sektordiagramme ja nende kombinatsioone. Kõik diagrammid võivad sisaldada mitmeid andmeseeriaid, võimaldades nii pakitust (*stacked*) kui ka kumulatiivsust (*cumulative*), mitme telje tuge ning automaatset telje skaleerimist. Samuti võimaldab Elycharts kasutada erinevaid interaktsioone nagu hiirjälgimist (*mouse tracking*) ja sündmusi (*events*), esile tõsta valitud kihte erinevate animatsioonidega, kohtspikreid, vihjeid jpm. Elycharts võimaldab kohandada diagrammi iga aspekti. Värvide, teksiparameetrite, suurused, joonestikutüübid on kergesti muudetavad. Elycharts toetab ka legendi loomist, andmete kohtspikreid, silte, malle ja palju muud. [16]

1.2.3 Google Charts

Google Charts on Google'i toode, mis sisaldab erinevaid diagramme kõige lihtsamatest joondiagrammidest kuni keerulisemate hierarhiliste puukaartideni (*treemaps*). [17] Google

Charts pakub kõikide diagrammide jaoks kasutusvalmis näiteid ning kergesti implementeeritavaid funktsioone.

Google Charts on avaldatud *JavaScript*'i klassidena ja põhineb HTML5'el ja SVG- ehk skaleeritavate vektorgraafikute tehnoloogial. Seetõttu on diagramme võimalik kuvada erinevates brauserites ja platvormidel, vajamata selleks eraldi brauserilisasid (*plug-ins*). Google Charts pakub 12 diagrammide põhitüüpi, mille hulka kuuluvad nt sektor-, punkt-, tulp-, joondiagrammid, tabelid jne. Ühtlasi on kasutajal võimalik luua interaktiivseid diagramme. Näiteks, kui kasutaja klikib hiirega mingil visuaalsel objektil, siis kuvab diagramm täpsemat infot selle objekti kohta. [18]

Sisemiselt talletab Google Charts oma andmeid andmetabelites. Kasutajad saavad luua erinevaid andmevaateid, et filtreerida tabelit. See pakub üsna suurt paindlikkust. Iga diagramm võib põhineda erineval andmevaatel, mis on viide olemasolevale andmetabelile. Läbi veebiteenuse on võimalik ühendada kliendipoolne andmetabel serveripoolse andmebaasiga. Kui andmebaas toetab Google Chart Tools Datasource protokollit, siis on kliendi poolisel programmil võimalik saata SQL-päringuid andmebaasi ning andmebaas tagastab vastavate andmetega täidetud andmetabeli. [18]

Positiivsed küljed [14]:

- Graafikuid on intuiitiivsed ja neid on kerge integreerida erinevatesse tehnoloogiatesse.
- Kohandatav ja kergesti kasutatav teek, et vastata peamistele vajadustele.
- Arusaadav ka mitte-tehnika-inimesele
- Graafikuid on võimalik kerge vaevaga kuvada .png formaadis

Negatiivsed küljed [14]:

- Puudub paindlikkus diagrammide tegemisel. Võimaldab ainult enimkasutatavaid diagramme luua – tulpdigramme, joondiagramme jms, mitte uut tüüpi diagramme.
- Mõningad Google Charts diagrammid ei toeta suuri andmehulki ning seega saab neid vaid piiratud andmehulga puhul rakendada.
- Kõikide Google Charts diagrammide puhul pole võimalik lisada erinevaid DOM funktsioone nagu suurendamine, klikkimine jms.
- Toetab ainult 2D diagramme.

1.2.4 SVGware

SVGware on *JavaScript*'i teek, mis võimaldab kergesti luua 2-dimensioonilisi interaktiivseid diagramme, tabelleid ja muid andmete visualiseerimise lahendusi. Teek toetub skaleeritavatele vektorgraafikutele (SVG). Lisaks sisseehitatud funktsionaalsustele nagu andmete ja skaalade joonestamine, suuruse muutmine (*resizing*), kerimine (*scrolling*), kerimisratta abil suurendamine (*scroll wheel zooming*) ning vasaku hiireklikiga suurendamine (*left-click zooming*), on toetatud ka mitmed uuesti esile kutsumised (*callbacks*), mis võimaldavad diagramme ulatuslikult kohandada, lisada interaktiivsust mitmel erineval viisil ning programmeerida keerulisemaid lahendusi. SVGware on disainitud nii, et diagrammid on vastuvõtlikud ka suuremate andmehulkade suhtes. See tähendab, et suure andmehulga puhul on nii diagrammi suurendamisel kui ka sellel kerides iga andmepunkt kättesaadav. [19]

Positiivsed küljed:

- Toetab suuri andmehulki, toimides sel puhul sama efektiivselt kui väiksemate andmehulkade puhul.
- Toetab sünkroonset suurendamist: kui on mitu graafikut ühel lehel.

2. Nõuded aruandlusmoodulile ja sobivaima teegi valimine

Käesolevas peatükis antakse täpsem ülevaade realiseeritavast aruandlusmoodulist ja sellele esitatud nõuetest. Seejärel võrreldakse eelnevalt käsitletud teekide võimalusi vastavalt aruandlusmoodulile esitatud nõuetele ning valitakse välja kaks sobivaimat teeki.

2.1 Senised lahendused

Seni ei ole liisinguettevõtte kasutanud ühtegi lahendust, et andmeid visuaalsel kujul graafikutel kuvada. Kogu vajalik info ja andmed on kätte saadud aja jooksul ning vastavalt vajadusele loodud aruannetest, kus info on esitatud tabelite ja arvudena. Tabelites olev info on küll spetsiifilisem ja täpsem ning sealt saab vajaduse korral osa infost kätte, kuid pideva ja kiire ülevaate saamine kõige olulisematest näitajatest tihti puudub. Siit tekibki vajadus aruandlusmooduli järele.

Tihti peale peab sügavamalt süvenema, et suures infohulgas märgata müüginumbrite mustreid ja seoseid ning äri arengut/taandarengut. Kujutades neid arve visuaalselt ning tuues olulisema info ühele lehele kokku, muudab see ülevaate saamise tunduvalt lihtsamaks ning olulise info kättesaamine ja järeluste tegemine ei nõua enam nii palju kallist aega ja üleliigset süvenemist. Seega optimeeriks aruandlusmoodul oluliselt müügiandmete analüüsi ja info omandamist. Kasutades osa info kujutamiseks diagramme, aitab see veelgi paremini näha andmete proportsioone. Oluline on numbritele anda just see visuaalne kuju, mis annaks kõige lihtsamini ja samal ajal kõige efektiivsemalt numbrite sisu edasi.

Liisinguettevõtte vajadus aruandlusmooduli järele tuli sellest, et teha müüginumbreid töötajatele, eelkõige projektijuhtidele igapäevaselt mugavalt kättesaadavaks, ülevaatlikuks ning kergesti mõistetavaks. Müüginumbrid peegeldavad nende jaoks tihti vajadust erinevatele partneritele ning müügikohtadele tähelepanu pööramiseks - nende täiendavaks koolitamiseks ning abistamiseks probleemide lahendamisel.

Vaadeldav aruandlusmoodul kujutab endast esimest osa aruannete visualiseerimise juures. Ettevõttel on tulevikus plaanis sarnaseid lahendusi luua ka teistes osakondades, mis tähendab, et oluline on aruandlusmooduli laiendatavus.

2.2 Metoodika sobivaima teegi leidmiseks

Analüüsimetoodika, mille tulemusena leitakse sobivaim teek aruandlusmooduli jaoks, hõlmab aruandlusmooduli nõuete kogumist ning nende nõuete põhjal eelnevalt valitud teekide võimaluste võrdlust ning hinnangut, kas või millisel määral on teegid võimelised aruandlusmooduli nõudeid täitma.

Teekide võrdlemiseks kasutatakse võrdlustabelit. Tabelis on esitatud paika pandud nõuetest tulenevad omadused ja funktsionaalsused, mida iga teegi puhul kontrollima hakatakse. Iga tabelis olev omadus on viidatud temaga seotud nõudele. Nõutud omaduste ja funktsionaalsuse olemasolu saab kontrollida teekide võimaluste testimise tulemusena. Teekide testimiseks kasutatakse nii dokumentatsioone, veebis kättesaadavaid näitelahendusi kui ka reaalselt katsetamist JSFiddle'i rakenduse abil (otse Internetis) ja erinevate diagrammide näitekoodide abil (autori oma arvutis).

Võrdluse tulemusena valib töö autor 2 kõige sobivamat teeki, mille abil loob aruandlusmooduli diagrammide võimalikult täpsed prototüübid, hinnates seejuures teekide kasutamise mugavust, lihtsust, koodiridade arvu ja aruandlusmooduli nõuete täitmist. Lõpuks peaks selguma sobivaim visualiseerimistEEK, mis oleks võimeline täitma ettevõtte poolt esitatud nõudeid ja andma seeläbi kõige paremini ja lihtsamini kasutajale infot edasi, nagu peatükis 1.1 välja toodud andmete visualiseerimise üldiste nõuete viimane punkt rõhutab.

2.3 Aruandlusmooduli nõuded

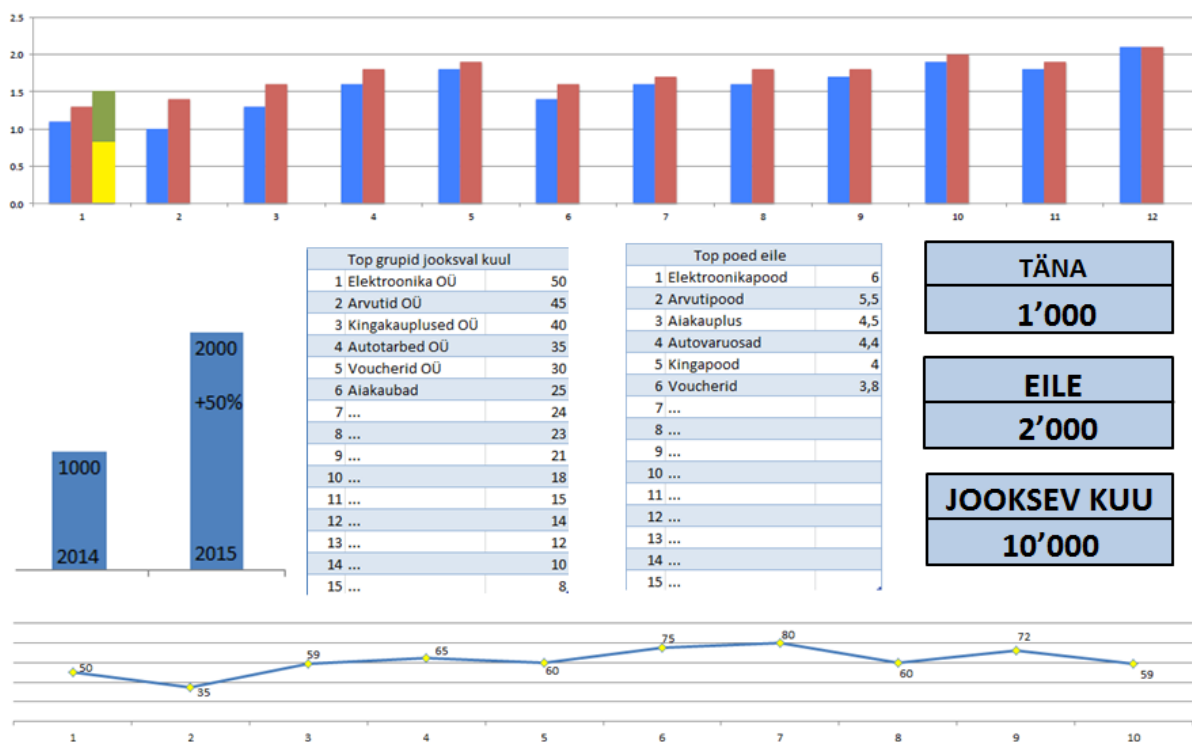
Järgnevalt tuuakse välja, milliseid andmeid soovitakse aruandlusmoodulil täpselt näha ja millisel kujul neid soovitakse esitada. Samuti antakse ülevaate erinevatest nõuetest aruandlusmoodulile, mille põhjal hakatakse edaspidi arenduseks võimalikke teeke omavahel võrdlema.

Olemasoleva ning igapäevaselt kasutuses olev liisingulepingute haldamise süsteem on üles ehitatud veebirakendusena. On kasutatud HTML-i, CSS-i, *JavaScript*'i, Pythonit ning kogu selle ümber Django veebirakenduse raamistikku, mis järgib model-view-controller ehk MVC arhitektuurimustrit. Andmebaasisüsteemina on kasutusel PostgreSQL. Üheks põhinõudeks

aruandlusmoodulile on selle ühilduvus liisingulepingute haldamise süsteemiga ehk aruandlusmooduli realiseerimiseks valitav teek peab olema võimeline toimima koos praeguse süsteemiga.

Ettevõtte poolt esitati konkreetsed nõuded, milliseid andmeid aruandlusmoodulis täpselt näha sooviti ning millisel kujul. Olulist rolli mängis ka paigutus, sest kõige tähtsam info sooviti kõige väljapaistvamana kuvada. Aruandlusmooduli üldpilt pidi välja nägema atraktiivne, aga mitte erinevate efektide läbi, vaid pigem sooviti näha lihtsust, korrektsust ja oma äri tunnusvärve. Samuti sooviti graafikuid kujutada 2D kujul, mitte ruumiliselt. Kogu pilti taheti kuvada eelkõige HD televiisoriekraanil, aga aruandlusmoodulile pidi olema ligipääs ka nende kasutatavast süsteemist läbi veebilehitseja.

Järgneval joonisel on kujutatud ettevõtte poolt esitatud visand aruandlusmoodulist (kasutatud värvid ja info on juhuslikud):



Joonis 2 Ettevõtte visand aruandlusmoodulist

Alapeatükis 1.1 oli andmete visualiseerimise ühe olulise üldise nõudena välja toodud just kindlaks määramine, mida üritatakse visualiseerida (numbreid, teksti jms.), mis tüüpi informatsiooniga on tegu (müügiandmetega) ja millisel kujul tahetakse seda informatsiooni edasi anda (eri tüüpi diagrammid, tabelid, numbrid). Järgnevalt esitatud loetelu soovitud

andmetest ja nende kujust näitab selgelt seda, et ettevõttel on juba endal selge nägemus ja nende poolt on juba kindlaks määratud, milliseid andmeid ja millisel kujul nad näha soovivad.

Nõuded soovitud andmete ja nende kuju kohta:

- 1) **Tänase päeva müük:** Suure numbrina kuvatud tänase päeva müük, täpsusega 1000 ühikut.
- 2) **Eilse päeva müük:** Suure numbrina kuvatud eilse päeva müük, täpsusega 1000 ühikut.
- 3) **Käesoleva kuu müük:** Suure numbrina kuvatud käesoleva kuu müük, täpsusega 1000 ühikut.
- 4) **Kuude müük:** Tulpdiagramm kuude müügist. Võrdluses eelmise aasta müük iga kuu kohta, selle aasta käesoleva kuu müügiprognoos, selle aasta müügiplaan iga kuu kohta ja jooksva aasta lõppenud kuude müük. Käesoleva kuu müügi tulp hakkab prognoositulpa vastavalt vähendama ning kui prognoos ületatakse, siis seda enam ei kuvata. Diagrammi y-teljel kuvada 5 skaalapunkti.
- 5) **Aastate müük:** Tulpdiagramm kahe tulbaga: jooksva aasta senini saavutatud müügisumma võrdlus eelmise aasta sama ajani saavutatud müügisummaga. Lisatud numbrid ja kui võimalik, siis ka protsent. Diagrammi telgesid mitte kuvada.
- 6) **TOP 10 grupid:** Tabel TOP 10 poegruppidest jooksva kuul ja vastavad müügisummad tuhandetes ühikutes, täpsusega 0,1 ühikut (nt kui on 1200, siis tabelis kuvatakse 1,2), sorteerida kahanevalt.
- 7) **TOP 10 poed:** Tabel TOP 10 poodidest eilsel kuupäeval, sorteerida kahanevalt.
- 8) **Viimase 15 päeva müük:** Joongraafik viimase 15 päeva müügist, iga päeva juures kuvada vastav müüginumber. Vertikaaltelge mitte kuvada.

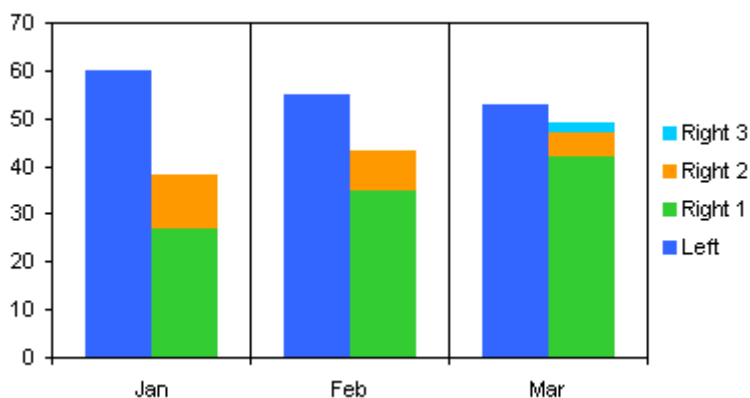
Nõuete täiendused vastavalt punktidele:

- 4) Selleks, et võrdlust iga kuu kohta arusaadavalt välja tuua, tuleks tulbad jagada kolmestesse rühmadesse:

- Esimene tulp grupis näitaks eelmise aasta müüki konkreetse kuu kohta.
- Teine tulp grupist näitaks käesoleva aasta plaani konkreetse kuu kohta.
- Kolmas tulp grupist tekiks jooksvalt vastavalt käesolevale kuule ning sisaldaks nii käesoleva kuu prognoosi kui ka käesoleva kuu müüki ühes tulpas, see tähendab, et kolmas tulp oleks nõ „pakitud“ (*stacked*) kokku kahest tulpast. Kolmas tulp tekib graafikule jooksvalt juurde vastavalt käesolevale kuule ning käesoleva kuu müük hakkaks aja möödudes vähendama kuu prognoosi.

Järeldus: Selline tulpdiaagramm peab võimaldama nii tulpade üksteise peale „pakkimist“ kui ka tulpade rühmitamist ehk tegemist on *clustered-stacked-column-chart*’iga.

Järgnevalt esitan selgituseks näite pakitud ja rühmitatud tulpadega diagrammist (*clustered-stacked-column-chart*):



Joonis 3 Pakitud ja rühmitatud tulpadega diagramm

Järgnevalt on ettevõtte poolt esitatud üldiste nõuete, soovitud arvude ning nende kuju nõuetest tuletatud ühtne nõuete loetelu, mille põhjal hakatakse edaspidi teete omavahel võrdlema.

Tuletatud nõuete puhul on silmas peetud nõuete 3 baasomadust [20]:

- 1) Ühene kontrollitavus - küsimusele „Kas nõue on täidetud?“ peab saama võimalikult üheselt vastata kas „jah“ või „ei“.
- 2) Kerge kontrollitavus – nõude kontroll ei tohi võtta palju aega.
- 3) Sõnastuse lihtsus ja lühidus – nõude sisutekst ei tohiks ületada 30 sõna.

Nõuded aruandlusmoodulile:

1. Aruandlusmoodulil kujutatavad diagrammid peavad olema 2-dimensioonilised.
2. Aruandlusmooduli diagramm kuude müügist peab olema kuvatud pakitud ja grupeeritud tulpadega diagrammil, mille y-teljel on 5 skaalapunkti.
3. Aruandlusmooduli diagramm aastate müügi võrdlusest peab olema kuvatud tulpdigrammina, ilma telgede ja skaaladeta (aastad, väärtused ja kasvuprotsent kuvatuna diagrammi peal)
4. Aruandlusmooduli diagramm viimase 15 päeva müügist peab olema kuvatud joondiagrammina, mille iga punkti juures on kuvatud väärtus ja vertikaaltelge pole kuvatud.
5. Aruandlusmoodulile peab olema ligipääs ettevõtte veebirakendusena töötavast süsteemist.
6. Aruandlusmoodulit peab olema võimalik kuvada erinevate resolutsioonidega ekraanidel (sh HD televiisoriekraanil)
7. Aruandlusmoodul peab olema efektiivselt kuvatav brauserites Chrome, Mozilla Firefox ja Safari
8. Aruandlusmooduli kujundust peavad läbima ettevõtte tunnusvärvid: tumesinine (#005488), sinine (#4B7794), valge (#FFFFFF), helekollane (#FFDD6D) ja oranž (#FF6600).
9. Aruandlusmoodul peab olema läbivalt eestikeelne.
10. Aruandlusmooduli kõik andmeplokid peavad olema korrektselt joondatud.
11. Aruandlusmoodulis kujutatud diagrammidele pole vaja legende juurde kuvada (kuna on piiratud hulk kasutajaid igapäevaselt).
12. Aruandlusmoodulit on võimalik laiendada ja edasi arendada.

2.4 Teekide võrdlus

Järgnevalt on esitatud tabel, kus tuuakse välja eelnevalt tutvustatud teekide võrdlus erinevate omaduste põhjal, mis tulenevad aruandlusmooduli nõuetest. Omaduste ja funktsionaalsuste olemasolu kontrollimiseks uuris töö autor erinevaid veebis saadaval olevaid näiteid, teekide dokumentatsioone ning püüdis erinevate diagramminäidete peal mõjutada nende objekte ja kujundust.

Tabelis on erinevate värvidega ära märgitud, millisel määral iga teek igat nõuet toetab. Roheline värv tähendab võrdlustabelis oleva nõude toetamist, kollane värv tähendab võrdlustabelis oleva nõude osalist toetamist ja punane värv nõude mittetoetamist. Kokku on nõudeid 14.

Tabel 2 Teekide võrdlus

	D3	Elycharts	Google Charts	SVGware
1. Toetab 2-dimensioonilisi diagramme (Nõue 1)	Jah	Jah	Jah	Jah
2. Toetab joondiagrammi (<i>line chart</i>) (Nõue 4)	Jah	Jah	Jah	Jah
3. Toetab tulpdiaagrammi (<i>column chart</i>) (Nõue 3)	Jah	Jah	Jah	Jah
4. Toetab korruga pakitud ja rühmitatud tulpadega diagrammi (<i>grouped-stacked column chart</i>). (Nõue 2)	Jah	Ei – ainult eraldi rühmitatud või pakitud tulpadega	Jah [21] – Material Column Chart	Võimalik, aga tulpade asetust peab täpselt koordinaatidega määrama
5. Üldine paindlik värvikasutus (Nõue 8)	Jah	Jah	Jah	Jah

	D3	Elycharts	Google Charts	SVGware
6. Paindlik värvikasutus korraga pakitud ja grupeeritud tulpadega diagrammil (Nõue 2 ja 8)	Jah – soovitud värvid saab määrata <code>ordinal.range([values])</code> abil, sisestades 'values' asemele soovitud värvid	Pakitud ja grupeeritud tulpadega diagrammi võimalus puudub	Ei lase pakitud ja grupeeritud tulpdigrammil kihtidele eraldi värvi määrata, vaid tekitab automaatselt ainult ühe põhivärvi erivarjundeid. [21]	Jah – nt: <pre>oStyle= { curve1: { bar: { rect: { fill:'Yellow' } } } } }</pre>
7. Paindlik ligipääs diagrammi telgedele (st lihtne eemaldada nii telgi kui skaalat) (Nõue 3 ja 4)	Jah – saab kasutada css atribuute, lisades neid diagrammi-elementide külge: <code>.css(„display“, „none“)</code>	Jah – skaala eemaldamiseks <code>defaultAxis: { labels: false }</code>	Jah: eraldi atribuuti telgede eemaldamiseks pole, aga teljed on võimalik taustaga samat värvi teha ja skaala eemaldamiseks määrata <code>textPosition: 'none'</code>	Ei – tundub, et see pole võimalik
8. Joon-diagrammi punktide väärtused (Nõue 4)	Jah – on võimalik punktid üles leida: <code>\$chart.find('.dot');</code> ja siis neile juurde kuvada soovitud väärtusi	Ei – kuvab punktide väärtusi ainult siis, kui hiirega punkti liikuda	Jah – andmetabeli deklareerimisel lisada veerg: <code>data.addColumn({type: 'number', role: 'annotation'})</code> [22]	Ei – ainult hiirega joonel liikudes näeb väärtuseid
9. Diagrammi-de horisontaal-telgede silte lihtne muuta (Nõue 9)	Jah	Jah – kasutades <code>labels</code> elemente	Jah	Silte keeruline muuta, saab ainult andmetest tulenevaid silte määrata
10. Diagrammide vertikaal-telgede skaalat lihtne muuta (Nõue 2)	Jah – muuta atribuuti <code>.ticks: var yAxis = d3.svg.axis().scale(y).ticks(5);</code>	Puudub info nii näidetest kui dokumentatsioonist	Ei – Material Charts puhul pole võimalik skaalat kohandada, vaid peab leppima automaatse skaalaga	Jah – nt <pre>scales: [{ scale: 'y1', items: [5, 10, 15] }]</pre>

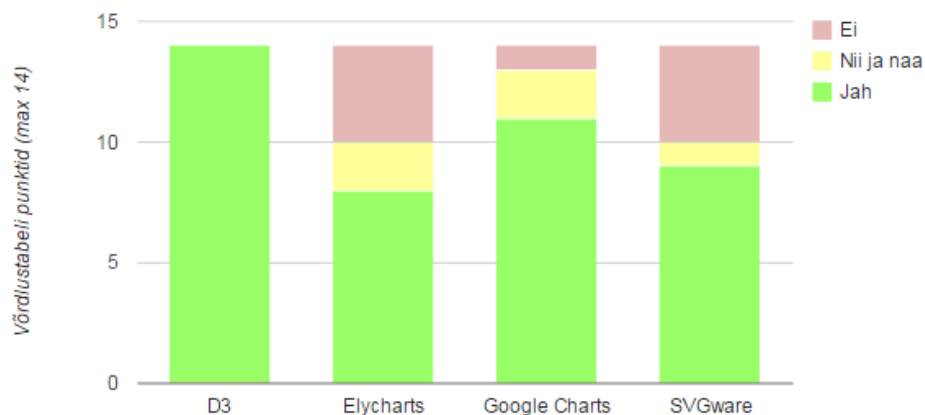
	D3	Elycharts	Google Charts	SVGware
11. Tulp-diagrammi tulpadel teksti kuvamine (Nõue 3)	Jah – <i>div</i> -elementide abil	Ei ole võimalik	Material Charts'ide puhul pole võimalik; klassikaliste tulpdigrammide puhul ainult väärtuste kuvamine.	Ei ole võimalik
12. Diagrammide legende võimalik eemaldada (Nõue 11)	Jah – legendi mitte tekitades	Puudub info diagrammide legendide kohta	Jah – <i>legend: 'none'</i> või <i>legend:{position: 'none'}</i>	Ei toetagi legendi
13. Toetab brausereid Google Chrome, Firefox ja Safari (Nõue 7)	Jah [12]	Jah [12]	Jah [12]	Jah [12]
14. Resolut- sioonist sõltumatu	Jah (SVG tehnoloogia)	Jah (SVG tehnoloogia)	Jah (SVG tehnoloogia)	Jah (SVG tehnoloogia)

Järgnevalt on esitatud eelneva võrdlustabeli tulemuste tabel, kus omaduste ja funktsionaalsuste toetus on iga teegi kohta välja toodud protsentidena ehk kui suure osa moodustasid täielikult toetatavad, kui suure osa osaliselt toetatavad ja kui suure osa mittetoetatavad teekide omadused ja funktsionaalsused kõikidest nõutest.

Tabel 3 Võrdlustabeli tulemused protsentidena

	D3	Elycharts	Google Charts	SVGware
Jah	100%	57%	79%	64%
Nii ja naa	0%	14%	14%	7%
Ei	0%	29%	7%	29%

Järgnevalt on kujutatud pakitud tulpdiagrammina võrdlustabeli tulemuste võrdlus, vastavalt sellele, kuidas iga teek on võimeline aruandlusmooduli nõudeid täitma. Siinkohal on selgelt näha, kuidas tulpdiagramm annab protsentide proportsioonidest palju selgema ülevaate.



Joonis 4 Võrdlustabeli tulemuste võrdlus

Järeldused iga võrreldud teegi kohta tabel 2 põhjal:

D3

D3 on kahtlemata väga paljude võimalustega teek. Lähtudes aruandlusmooduli nõuetest, on kõik nõudmised täidetavad ja disainivõimalused piiramatud. Disaini poolelt annabki väga palju paindlikkust juurde just võimalus siduda D3 elemente CSS-iga.

Üks põhilisemaid miinuseid D3 juures on keerulisus, mis tähendab pikemat õppimisprotsessi. Seda aga eelkõige nendele, kellel on vaja rohkem ja pidevalt andmete visualiseerimisega tegeleda. Keerulisemat õppimisprotsessi kompenseerib aga teadmine, et veebis kättesaadav näidetepagas on tänu suurele kasutajaskonnale muljetavaldavalt suur, seega väga palju saab ära teha just näidete baasil. Lisaks sellele võimaldab D3 ligipääsu igale objektile ning suhelda *div*-elementide abil teiste elementidega lehel, mis teevad samuti objektide kujundamise ja täiendamise väga paindlikuks.

Elycharts

Elycharts on üsna lihtne teek, mis pakub visuaalselt atraktiivseid diagramme, mida on kerge realiseerida. Tema võimalused jäävad aga kohati veidi piiratuks. Näiteks ei ole võimalik Elycharts'iga luua samaaegselt grupeeritud ja pakitud tulpadega diagrammi. Elycharts ei

võimalda tulpdiagrammi tulpadele ega joondiagrammi punktidele väärtusi või teksti juurde kuvada, vaid võimaldab ainult hiirega punktile või tulpale peale liikudes kuvada selle väärtust.

Lisaks sellele on Elycharts'i dokumentatsioon ja näidetepagas pigem kesine, mis suurendab teadmatust, milleks teek võimeline on.

Google Charts

Google Charts paistis silma lihtsusega. Seda on lihtne kasutada ja graafikute loomiseks piisas üsna lühikesest koodist. Lihtne oli lisada joondiagrammi punktidele siltidena väärtusi juurde. Selleks piisas vaid ühe andmeveeru lisamisest (*'annotation'*) ja ridades vastavale annotatsiooniveerule väärtuste andmisest.

Siinkohal peab aga vahet tegema „Classic“ diagrammide ja „Material“ diagrammide vahel. Mitmed funktsioonid, mida „Classic“ diagrammid võimaldavad, on „Material“ diagrammidel veel puudu. Aruandlusmooduli nõuetest lähtudes pole „Classic“ võimeline looma samaaegselt rühmitatud ja pakitud tulpadega diagrammi. „Material“ ei võimalda aga diagrammide skaalat muuta ega tulpadel vastavaid väärtusi kuvada. Kui aga mõlemaid – nii „Classic“ kui „Material“ – diagramme kasutada, siis kaob aruandlusmoodulil ühtne stiil, sest „Material“ tulpdiagrammi tulbad on ümarate nurkadega, „Classic“ diagrammid aga teravate nurkadega. Veel üks miinus Google Charts puhul seisnes samaaegselt grupeeritud ja pakitud tulpadega diagrammi värvivalikus. Grupeeritud tulpadele sai küll eraldi värve määrata, kuid iga tulba eraldi kihile eraldi värvi määrata ei saanud, vaid ülemised kihid lähtusid terve tulba põhivärvist ja väljendusid sama värvi üha heledamates toonides. Aruandlusmooduli nõuetest võib aga selgelt välja lugeda, et on nõutud ettevõtte tunnusvärvid.

SVGware

SVGware on andmete haldamises väga võimekas, see tähendab, et toimib hästi suurte andmehulkadega, kaotamata sealjuures jõudlust. Samas on selle teegi kujundusvõimalused üsna piiratud. Diagrammid näevad üsna matemaatilised välja ja erinevate objektide muutmine on raskendatud või üldse võimatu. Samuti peab andmete kuvamiseks määrama koordinaatidega ära ka kuvatava andmeobjekti täpse asukoha diagrammil. SVGware võimaldab luua samaaegselt rühmitatud ja pakitud tulpadega diagrammi, kuid väärtusi ja teksti tulpadel kuvada polnud võimalik. Ka joondiagrammil polnud võimalik iga punkti

väärtust eraldi kuvada, see oli võimalik vaid hiirega punktile liikudes. Diagrammi telgedele puudus samuti ligipääs, st neid polnud võimalik eemaldada ega muuta.

SVGware sobib hästi erinevate statistiliste andmete kuvamiseks, kus on vaja palju andmeid näidata ja võimalikult täpselt. Antud olukorras on aga suur roll just visuaalselt atraktiivsusel, mistõttu on oluline paindlik ligipääs diagrammi elementidele ja kujundusvõimalustele. Samuti on piiratud SVGware diagrammide info edasi andmine HD televiisoriekraanil (nõue 6), kus aruandlusmoodul peaks andma võimalikult palju informatsiooni juba peale vaadates. Kuna SVGware ei võimalda diagrammidel vastavaid väärtuseid juurde kuvada, vaid näitab neid ainult hiirega elementide liikudes, siis televiisoriekraanil kuvades jääb palju infot saamata.

Otsus

Kõige paremini täitsid aruandlusmooduli nõudeid Google Charts ja D3 teegid. D3 oli võimeline kõiki aruandlusmooduli nõudeid täitma. Google Charts ei toetanud skaala kohandamist ning toetas osaliselt annotatsioone ja samaaegselt pakitud-rühmitatud tulpadega diagrammi paindlikkust värvivaliku puhul.

Edasiseks prototüüpide realiseerimiseks osutusid valituks Google Charts ja D3 teegid.

3. Aruandlusmooduli osade prototüübid

Selles peatükis esitatakse kahe sobivaima teegi abil loodud aruandlusmooduli diagrammide prototüübid, analüüsitakse neid ja tehakse järeldused.

3.1 Prototüübid ja nende analüüs

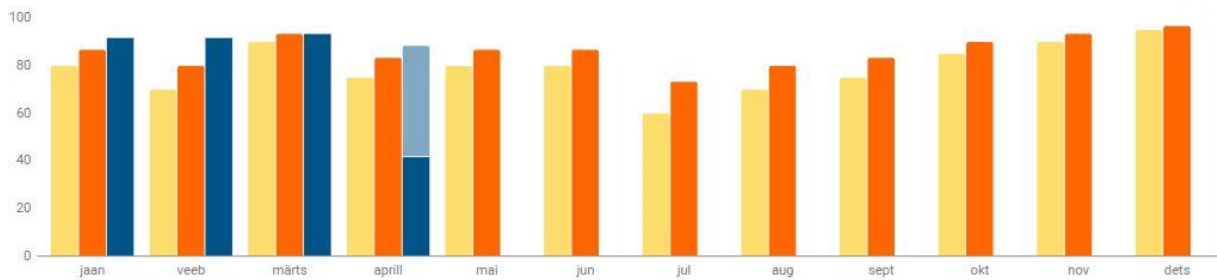
Aruandlusmoodulis oli nõutud 3 erinevat diagrammi kujul aruannet: tulpdiagramm kuude müügist, joondiagramm viimase 15 päeva müügist ja tulpdiagramm aastate müügist. Töö autor proovis eelnevalt valituks osutunud teekide abil luua nende diagrammide prototüübid, mis vastaksid võimalikult täpselt aruandlusmooduli nõuetele.

Prototüüpide loomisel on lähtutud peatükis 1.1 joonisel 1 välja toodud andmete visualiseerimise põhisammudest – näidisandmete kogumine, nende numbriline teisendus (ühendatud tähenduse andmine jms), andmeanalüüs (andmete rühitamine ja skaleerimine), graafiline esitus (diagrammide loomine) ja kasutaja/arendaja tegevus (diagrammide elementide ja kujunduse kohandamine jms).

Tulpdiagramm kuude müügist

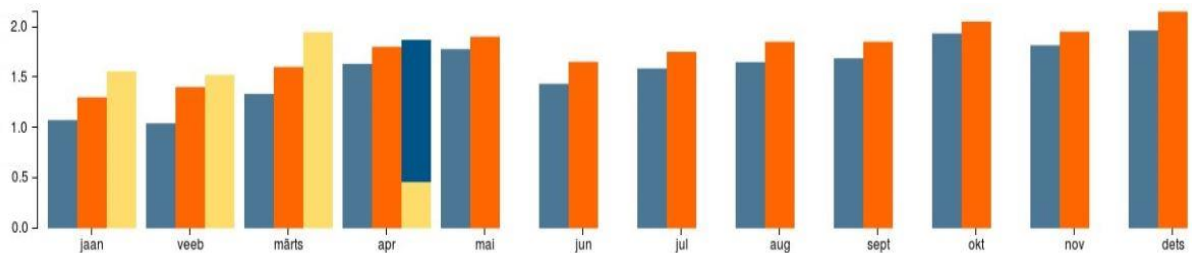
Kuude müügi võrdluse tulpdiagrammi puhul oli nõutud, et tegemist oleks samaaegselt rühmitatud ja pakitud tulpadega diagrammiga, vertikaalteljel oleks soovitatavalt 5 skaalapunkti ning kasutatud oleks ettevõtte tunnusvärve.

Järgnevalt on esitatud Google Charts teegiga loodud tulpidiagrammi prototüüp kuude müügi võrdluse esitlemiseks.



Joonis 5 Tulpdiagramm kuude müügist (Google Charts)

Järgnevalt on esitatud D3 teegiga loodud tulpdiagrammi prototüüp kuude müügi võrdluse esitlemiseks:



Joonis 6 Tulpdiagramm kuude müügist (D3)

Tähelepanekud kuude müügi tulpdiagrammi prototüübi loomisel:

Tabel 4 Tähelepanekud kuude müügi tulpdiagrammide prototüüpidest

Google Charts	D3
<ul style="list-style-type: none"> • Koodiridade arv ~49 • Üsna kiire ja lihtne luua • Pakitud tulpadele ei saa eraldi värve määrata (ainult üks põhivärv) • Vertikaaltelje skaalat ei saa ise modifitseerida, skaala sõltub automaatselt diagrammi kõrgusest • Horisontaaltelje silte lihtne muuta 	<ul style="list-style-type: none"> • Koodiridade arv ~88 • Üsna aeganõudev loomine • Paindlik värvide määramine pakitud tulba erinevatele kihtidele • Vertikaaltelje skaalapunkte lihtne muuta • Horisontaaltelje silte lihtne muuta • Andmeid saab ette anda .csv-failina

Joondiagramm viimase 15 päeva müügist

Joondiagrammi puhul oli nõutud, et joone iga punkti juures oleks kuvatud vastav väärtus ning joondiagrammi vertikaaltelge ei kuvataks, kuna kogu info saadakse efektiivsemalt kätte otse punktide väärtustest.

Järgnevalt on esitatud Google Charts teegi abil loodud joondiagrammi prototüüp viimase 15 päeva müügist:



Joonis 7 Joondiagramm viimase 15 päeva müügist (Google Charts)

Järgnevalt on esitatud D3 teegi abil loodud joondiagrammi prototüübi viimase 15 päeva müügist:



Joonis 8 Joondiagramm viimase 15 päeva müügist (D3)

Tähelepanekud viimase 15 päeva müügi joondiagrammi prototüübist:

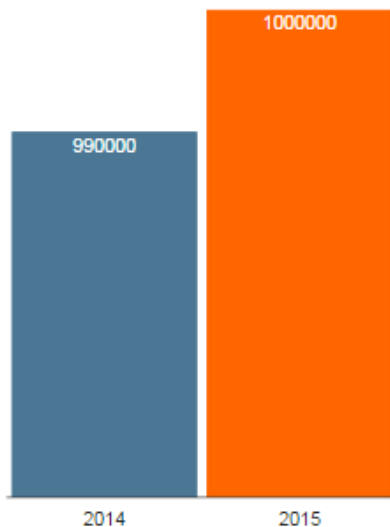
Tabel 5 Tähelepanekud viimase 15 päeva müügi joondiagrammi prototüübist

Google Charts	D3
<ul style="list-style-type: none">• Koodiridade arv ~28• Lihtne kuvada punktidele väärtusi (kõigest 1 koodireaga)• Diagrammi vertikaaltelge lihtne eemaldada• Joondiagrammi ala ei saanud värviga ära täita – selleks tuleks kasutada Area Charti, kuid seal ei saa punktidele annotatsioone lisada• Punktide väärtuste asukohad on automaatsed	<ul style="list-style-type: none">• Koodiridade arv ~45• Punktidele väärtuste juurde kuvamine nõudis oodatust kauem aega• Punktide väärtuste kuvamine nõudis oodatust rohkem koodiridu• Diagrammi vertikaaltelge lihtne eemaldada• Joondiagrammi ala sai värviga täita• Punktide väärtuste asukohad sai ise määrata (et nt kõik asuksid joone peal)

Tulpdiagramm aastate müügist

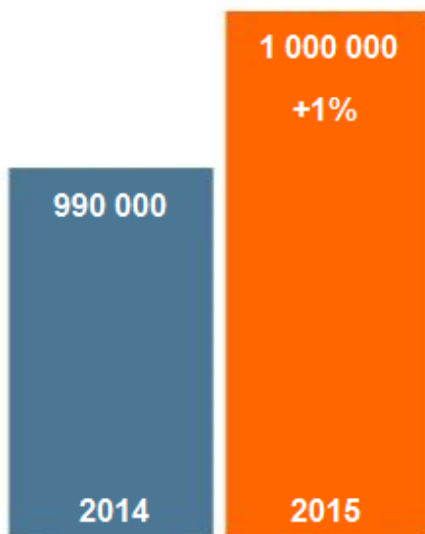
Aastate müügi võrdluse tulpdiagrammi puhul nõuti ilma telgedeta kahetulbalist diagrammi, kujutades sealjuures nii aasta kui ka müügisumma tulba peal ning võimalusel lisades ka muutuseprotsendi, mis näitaks müügisummade kasvu või kahanemist, võrreldes eelmise aasta müügisummaga praeguse ajani.

Järgnevalt on esitatud Google Charts abil loodud aastate müügi võrdluse tulpdiagrammi prototüüp:



Joonis 9 Tulpdiagramm aastate müügist (Google Charts)

Järgnevalt on esitatud D3 teegi abil loodud aastate müügi võrdluse tulpdiagrammi prototüüp:



Joonis 10 Tulpdiagramm aastate müügist (D3)

Tähelepanekud viimase 15 päeva müügi joondiagrammi prototüübist:

Tabel 6 Tähelepanekud aastate müügi tulpdiagrammi prototüübist

Google Charts	D3
<ul style="list-style-type: none">• Koodiridade arv ~37• Tegemist „Classic“ tüüpi diagrammiga, st tuld on väljanägemiselt veidi erinevad (kandilised) kui Material“ tüüpi kuude müügi tulpdiagrammi tuld (ümarate nurkadega)• Vertikaaltelge kerge eemaldada• Horisontaaltelje pidi alles jätma, sest vastavaid silte ei saanud kuvada tuld peal• Tulbal sai kuvada ainult ühte annotatsiooni korraga, st kasvuprotsenti ei saanud lisada.• Tulpadele sai eri värve määrata.	<ul style="list-style-type: none">• Koodiridade arv ~53• Nii vertikaal- kui ka horisontaaltelg eemaldatavad• Tulpade väärtusi, horisontaaltelje silte ja kasvuprotsenti sai kuvada tuld peal (<i>div</i>-elementide abil)• Tuld peal kuvatava teksti asukohta ja stiili sai CSSi abil sobivaks muuta.• Tulpadele sai eri värve määrata.

3.2 Järeldused

Eelnevalt loodi kahe omadustelt ja funktsionaalsustelt sobivaima teegiga aruandlusmooduli diagrammide prototüübid. Tulemused olid üsna sarnased, kuid esines ka mõningaid erinevusi.

D3 teek oli võimeline kõiki aruandlusmooduli diagrammidele esitatud nõudmisi täitma, kuid prototüüpide lahendused nõudsid mõnevõrra mahulisemat koodi ja rohkem aega. See on tingitud sellest, et D3 paindlikkus ja diagrammi elementidele ligipääsetavus on peaaegu täielikult tagatud ja automaatseid lahendusi ja funktsionaalsust on pigem vähe. See annab arendajale võimaluse luua üsna spetsiifilistele nõuetele vastavaid diagramme, kus iga detail on oluline ja läbi mõeldud. Samas tuleks seejuures kindlasti arvestada ka arendaja kogemuse ja õpivõimega, sest sellest sõltub, kui kiiresti on ta võimeline töötava lahenduseni jõudma.

Google Charts näitas samuti oma võimet saada peaaegu täielikult hakkama aruandlusmooduli nõuete täitmisega. Tähelepanuväärne on Google Charts teegi lihtsus – see võimaldas väga mugavalt, üsna kiiresti ja pigem väikese koodimahuga diagramme luua. Erinevate omaduste ja funktsionaalsuste uurimisel aga selgus, et Google Charts kasutab 2 erinevat tüüpi diagramme – „Classic“ ja „Material“ – mis on nii oma vaikimisi kujunduse kui ka võimaluste poolest veidi erinevad. Mõningad funktsionaalsused, mida „Classic“ tüüpi diagrammid toetavad, pole „Material“ tüüpi diagrammide puhul võimalik kaasata, nagu näiteks annotatsioonide (*annotations*) kasutamine, mis võimaldab tulpdiagrammide tulpadele ja joondiagrammi punktidele vastavaid väärtusi juurde kuvada. Lisaks sellele on „Material“ tüüpi tulpdiagrammidel vaikimisi ümarate nurkadega tulbad, mille kujundust ei saa muuta. See mängib rolli siis, kui ühele lehele on vaja kuvada „Material“ tüüpi tulpdiagramm (mis võimaldab tulpi samaaegselt rühmitada ja pakkida) ja „Classic“ tüüpi tulpdiagramm (mis võimaldab annotatsioone kasutada) ja see omakorda mõjub negatiivselt aruandlusmooduli ühtsele kujundusele.

Lõpliku otsuse tegemisel tuleks lähtuda sellest, mida ikkagi peetakse aruandlusmooduli juures kõige olulisemaks – kas võimalikult kiiret ja lihtsat lahendust või nõuete 100%-list täitmist ja edasist paindlikku laiendamisvõimalust – ning kuivõrd mängib rolli koodimaht ja ajakulu. Tuleb arvestada ka sellega, et käesoleva aruandlusmooduli puhul ei ole tegu ühekordse projektiga, vaid tulevikus plaanitakse sarnaseid lahendusi ka ettevõtte teistesse osakondadesse, seega tuleb osata ette mõelda, kas edasised lahendused nõuavad palju kujunduslikku paindlikkust (D3) või ollakse nõus leppima konkreetse teegi pakutud mõnevõrra piiratud, kuid lihtsasti ja kiiresti realiseeritavate võimalustega (Google Charts).

Töö autori arvates peaks valima pigem paindlikumate võimalustega teegi ehk D3. Olgugi et see nõuab rohkem süvenemist, pühendumist ja aega, on see võimeline kõiki nõudeid täitma ja seda eeldatavasti ka tulevikus. Kui aga ettevõtte otsustab kiirema ja lihtsama lahenduse ehk Google Charts teegi kasuks, loobudes mõningatest nõuetest või aktsepteerides nende alternatiive praeguse aruandlusmooduli puhul, peaks ta sellega arvestama ka edaspidi, et kujunduslike nõuetega ei saa liiga spetsiifiliseks minna.

Kokkuvõte

Ettevõttel tekkis vajadus oma müügiandmeid visualiseerida, kuna see aitab ettevõtte töötajatel andmeid paremini mõista ja analüüsida. Selleks anti ülesanne luua ettevõttele aruandlusmoodul vastavalt nende poolt esitatud nõuetele ja soovidele. Aruandlusmooduli loomiseks oli vaja leida sobivaim *JavaScript*'i teek, aga kuna tänapäeval on *JavaScript*'i andmete visualiseerimise teekide valik suur ja nende võimalused on mõnevõrra erinevad, siis tuli kõigepealt analüüsida ja võrrelda erinevaid andmete visualiseerimise teeke ja selle põhjal valida aruandlusmooduli loomiseks neist sobivaim.

Käesoleva töö eesmärgiks oli leida sobivaim *JavaScript*'i andmete visualiseerimise teek ettevõtte aruandlusmooduli loomiseks.

Töö eesmärgi saavutamiseks tutvus töö autor andmete visualiseerimise teemaga üldiselt ja andis ülevaate neljast *JavaScript*'i SVG-tehnoloogial põhinevast andmete visualiseerimise teegist: D3, Elycharts, Google Charts ja SVGware. Seejärel tutvustas autor aruandlusmooduli nõudeid ning tuletas nendest omadused ja funktsionaalsused, mille põhjal eelnevalt tutvustatud teeke omavahel võrdlema hakata. Teekide võrdlemiseks koostas autor võrdlustabeli, kus märkis ära, milline teek millist omadust või funktsionaalsust toetab või ei toeta ja tegi sellest järeldused. Võrdluse tulemusena valis töö autor välja 2 sobivaimat teeki: Google Charts ja D3, millega hakkas realiseerima aruandlusmooduli diagrammide prototüüpe. Loodud prototüüpide põhjal tegi töö autor järeldused kummagi teegi kohta.

Töö tulemusena leidis autor, et kui ettevõtte soovib kiiret ja lihtsat lahendust ning on nõus selle arvelt loobuma mõnest kujunduslikust nõudest, siis on sobivaim teek aruandlusmooduli jaoks Google Charts. Kui aga ettevõtte soovib kõikide aruandlusmooduli nõuete 100%-list täitmist ja kujunduslikku paindlikkust tulevikus realiseeritavate sarnaste lahenduste puhul ning on nõus veidi suurema ajakulu ja koodimahuga, siis on sobivaim teek aruandlusmooduli jaoks D3.

Töö edasiarendusena võib kaaluda sarnase sobivaima visualiseerimisteegi valimise meetodi kasutamist ka teistsuguste andmete visualiseerimise ülesannete puhul, mis ei ole ainult müügiandmetega seotud.

Summary

A company had a need to visualize their sales data, because it helps employees to understand and analyse data better. To meet the need, the company gave a task to create a reporting module for them, which would grant the requirements and wishes of the company. For creating the reporting module, it was necessary to find the most suitable *JavaScript* library, but in today's world there is a great choice of different *JavaScript* data visualization libraries. So first of all, it was important to analyse and compare different data visualization libraries and on the basis of the comparison choose the most suitable library for the reporting module.

The aim of current thesis was to find the most suitable *JavaScript* data visualization library for creating a company's reporting module.

To achieve the aim, the author of this thesis gained some knowledge of data visualization subject in general and gave insight of four different *JavaScript* data visualization libraries, which are based on SVG-technology: D3, Elycharts, Google Charts and SVGware. Then the author introduced the requirements for the reporting module and derived features and functionalities from them, on the basis of which the author started to compare previously introduced libraries. The author composed a comparison table, where she marked out, which library supports which feature or functionality and made a conclusion out of it. As a result of the comparison, the author choosed two of the most suitable libraries: Google Charts and D3, and started to create the prototypes of the reporting module with them. On the basis of these prototypes, the author made final conclusions for both libraries.

As a result of current thesis, the author found that if the company needs a quick and simple solution and is willing to give up some requirements of design features, then the most suitable library for the reporting module would be Google Charts. However, if the company wants all the requirements to be fulfilled, styling flexibility at similar solutions created in the future and is willing to accept slightly increased time expenditure and code capacity, then the most suitable library for the reporting module would be D3.

For further development, it might come under consideration to use similar method of choosing the most suitable data visualization library in other data visualization tasks as well, not only in visualizing sales data.

Kasutatud kirjandus

1. Saito T., Ouyang J. Client-Side Data Visualization. [WWW]
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5211550> (06.04.2015)
2. Ward M. Overview of Data Visualization. [WWW]
<http://web.cs.wpi.edu/~matt/courses/cs563/talks/datavis.html> (07.04.2015)
3. SAS The Power To Know: Data Visualization. What Is It And Why It Is Important. [WWW]
http://www.sas.com/en_us/insights/big-data/data-visualization.html (07.04.2015)
4. Kamran Parsaye, Mark Chignell, "Intelligent Database Tools & Applications". (QA 76.9 Dbm.PS) (10.04.2015)
5. W3Schools: HTML5 SVG. [WWW]
http://www.w3schools.com/html/html5_svg.asp (10.04.2015)
6. W3Schools: HTML5 Canvas. [WWW]
http://www.w3schools.com/html/html5_canvas.asp (10.04.2015)
7. Sucas M. SVG or Canvas? Choosing Between The Two. [WWW]
<https://dev.opera.com/articles/svg-or-canvas-choose/> (10.04.2015)
8. Tutorialspoint: JavaScript Overview [WWW]
http://www.tutorialspoint.com/javascript/javascript_overview.htm (12.04.2015)
9. Oracle: Chapter 1: JavaScript Overview [WWW]
<http://docs.oracle.com/cd/E19957-01/816-6409-10/intro.htm> (12.04.2015)
10. W3Schools: JavaScript Tutorial [WWW]
<http://www.w3schools.com/JS/default.asp> (12.04.2015)
11. TIOBE Software: TIOBE Index for March 2015. [WWW]
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> (17.04.2015)
12. SocialCompare: Javascript Graphs and Charts libraries. [WWW]
<http://socialcompare.com/en/comparison/javascript-graphs-and-charts-libraries> (17.04.2015)
13. Maclean M. D3 Tips and Tricks: What is d3.js? [WWW]
<https://leanpub.com/D3-Tips-and-Tricks/read> (20.04.2015)
14. Sumanth K. Advantages of Google Charts over D3.js. [WWW]
<http://blog.socialcops.org/engineering/d3-js-versus-google-charts> (20.04.2015)

15. Grehan R. Review: 4 supercool JavaScript Tools for Data Visualization. [WWW]
<http://www.infoworld.com/article/2612006/application-development/article.html>
(20.04.2015)
16. elycharts.com: Features [WWW]
<http://elycharts.com/features> (22.04.2015)
17. Google Developers: Using Google Charts [WWW]
<https://developers.google.com/chart/interactive/docs/index> (23.04.2015)
18. Ying Z. Introducing Google Chart Tools and Google Maps API in Data Visualization Courses (Ying Zhu, Georgia State University) [WWW]
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6353424> (23.04.2015)
19. svgware.com: Interactive Charts For The Web [WWW]
<http://svgware.com/> (25.04.2015)
20. Potter P. Süsteemi nõuete esiletoomine ja analüüs [WWW]
<http://maurus.ttu.ee/sts/wp-content/uploads/2011/10/S%C3%BCsteemi-n%C3%B5uete-esiletoomine-ja-anal%C3%BC%C3%BCs.pdf> (25.04.2015)
21. JSFiddle: Google Charts grouped-stacked column chart example [WWW]
<http://jsfiddle.net/p7o0pjgg/> (27.04.2015)
22. Google Charts: DataTable Roles [WWW]
<https://developers.google.com/chart/interactive/docs/roles> (05.05.2015)