

Tallinna Ülikool
Informaatika Instituut

FOXPRO GENEETIKAALASE PÄRANDRAKENDUSE PORTIMINE VEEBIPLATVORMILE

Bakalaureusetöö

Autor: Lauri Elias
Juhendaja: Jaagup Kippar

Autor: „ 2014
Juhendaja: „ 2014
Instituudi direktor:..... „ 2014

Tallinn 2014

Autorideklaratsioon

Deklareerin, et käesolev bakalaureusetöö on minu iseseisva töö tulemus ning seda pole varem kaitsmisele esitatud.

Kuupäev:..... Autor:

Sisukord

| | |
|--|----|
| Sissejuhatus..... | 5 |
| 1. Pärandrakenduse puudused | 6 |
| 1.1 Tarkvaratõe lõpp | 6 |
| 1.2 Nähtavad tulevikuraskused taskukohase arenduse tellimisel..... | 6 |
| 1.3 Windows'i töölaarakenduse eripärad | 7 |
| 1.4 Võrdlemisi keeruline kasutajaliides ja paigaldusprotsess..... | 7 |
| 1.5 Halvas seisukorras andmebaas | 9 |
| 2. Arenduse ettevalmistus ja planeerimine | 10 |
| 2.1 Andmete eksport VFP rakendusest | 10 |
| 2.2 Nõuete väljaselgitamine | 10 |
| 3. Nõuded..... | 11 |
| 4. Arhitektuur | 13 |
| 5. Arendusvahendite valik..... | 16 |
| 5.1 PHP..... | 16 |
| 5.2 Symfony 2.3..... | 16 |
| 5.3 FOSRestBundle..... | 17 |
| 5.4 MySQL | 17 |
| 5.5 AngularJS..... | 17 |
| 5.6 Bootstrap | 17 |
| 5.7 GitHub | 18 |
| 6. Arenduse käik..... | 19 |
| 7. Testimine | 20 |
| 7.1 Testimisprotseduurid | 20 |
| 7.2 Nõuetele vastavus | 20 |
| 8. Paigaldus | 22 |
| 8.1 Symfony projektikausta loomine | 22 |
| 8.2 Projektifailide hankimine GitHub'ist..... | 22 |

| | | |
|-----|--|----|
| 8.3 | Serveri juurkausta suunamine Symfony failide pihta | 23 |
| 8.4 | Seadistuste kontrollimine | 23 |
| 8.5 | Lisasõltuvuste paigaldamine | 23 |
| 8.6 | Andmebaasi paigaldamine | 23 |
| | Kokkuvõte | 24 |
| | Summary..... | 25 |
| | Viited..... | 26 |
| | Lisad | 28 |
| | Lisa 1..... | 28 |
| | Lisa 2..... | 28 |
| | Lisa 3..... | 28 |

Sissejuhatus

Töö teema oli välja pakutud Tallinna Ülikooli üliõpilaste akadeemiliste tööde teemade registris. Autorile tundus see piisavalt huvitava valikuna, peale selle oli teema süsteemi sisestanud autorile sobiv juhendaja ja näis, et lisaks akadeemilisele oleks valmival töö ka praktiline väljund, mis säilitab mingigi kasulikkuse ka pärast kaitsmiskomisjoni ees esinemist. Autorit meelitas mõte, et tema töö on ehk „pildil“ pikemaks ajaks, kui akadeemilises maailmas üldiselt kombeks, kus, nagu Indiana Ülikooli uuringust selgub, 90% töödele ei viidata kunagi ning 50% ei loe mitte keegi peale selle autori, hindajate ja toimetaja. (Meho, 2007) Lisaks paelus autorit mõte tunda õppida kunagi suurt populaarsust nautinud pärandvara millest enne kuulnudki polnud.

Andmed, mille parem kättesaadavus siinse töö eesmärgiks on, töö idee ja ka mõningal määral selle teostus pärinevad Ilhama Mirzaliyeva'lt, kes oli töö kirjutamise ajal Tallinna Ülikooli doktorant. Teda tutvustas autorile Jaagup Kippar.

Arenduse sisendiks on aegunud töölaarakendus, väljundiks kergesti hallatav ning edasiarendatav, esialgu inglise- ja aserbaidžaanikeelne veebirakendus, mille abil on võimalik otsida, sorteerida ja eri formaatides eksportida geneetikaalaseid andmeid. Täpsemalt on tegu andmetega taimede geneetilise materjali hoiustamise kohta. Töö kirjutamise ajal ei kuulu arenduse skooopi näiteks põhjaliku haldusmooduli, kasutajaõiguste süsteemi ja palju muu arendamine, andmete muutmise on esialgu kergesti teostatav kasvõi PHPMyAdmin tarkvara kaudu, eriti, kuna haldamisega hakkab tegelema peamiselt 1 isik, kelle hakkamasaamises ei ole põhjust kahelda kasvõi seetõttu, et tegu on tehnikateaduste doktorandiga. Eelnimetatud funktsionaalsuse arenduseni on küll plaanis jõuda pärast töö esitamist.

Töö teostamiseks on autor valinud järgmised vahendid: PHP, MySQL, Symfony 2.3, AngularJS, FOSRestBundle, Apache2, Bootstrap ning Github.

1. Pärandrakenduse puudused

1.1 Tarkvaratõe lõpp

Esimene ja tõenäoliselt kõige silmatorkavam probleem vana rakenduse juures on, et tegu on Visual FoxPro, edaspidi „VFP“, täpsemalt selle versiooni 5.0 rakendusega. VFP 5 nägi ilmavalgust 1997. aastal ja selle programmeerimisekeele ning andmebaasihalduri hiilgeajad jäädvadi suuresti samasse kümnendisse. (FoxPro, 2014) Algselt oli projekti eesmärgiks pärandrakenduse portimine uuemale VFP versioonile ja sellele näiteks ActiveVFP tehnoloogia abil veebiliidese loomine, kuid siinses peatükis väljatoodud põhjustel tuli plaane muuta. Microsoft lõpetab VFP toe 13. jaanuaril 2015 (Microsoft, 2014), seegi tähtaeg kehtib ainult VFP 9.0 kohta väljalaskeastaga 2007, mis tähendaks, et rakendus oleks vähem kui aasta pärast uuenduskuuri läbimist taas peaaegu et lootusetult aegunud. Miks „peaaegu“, selgub järgnevas lõigus.

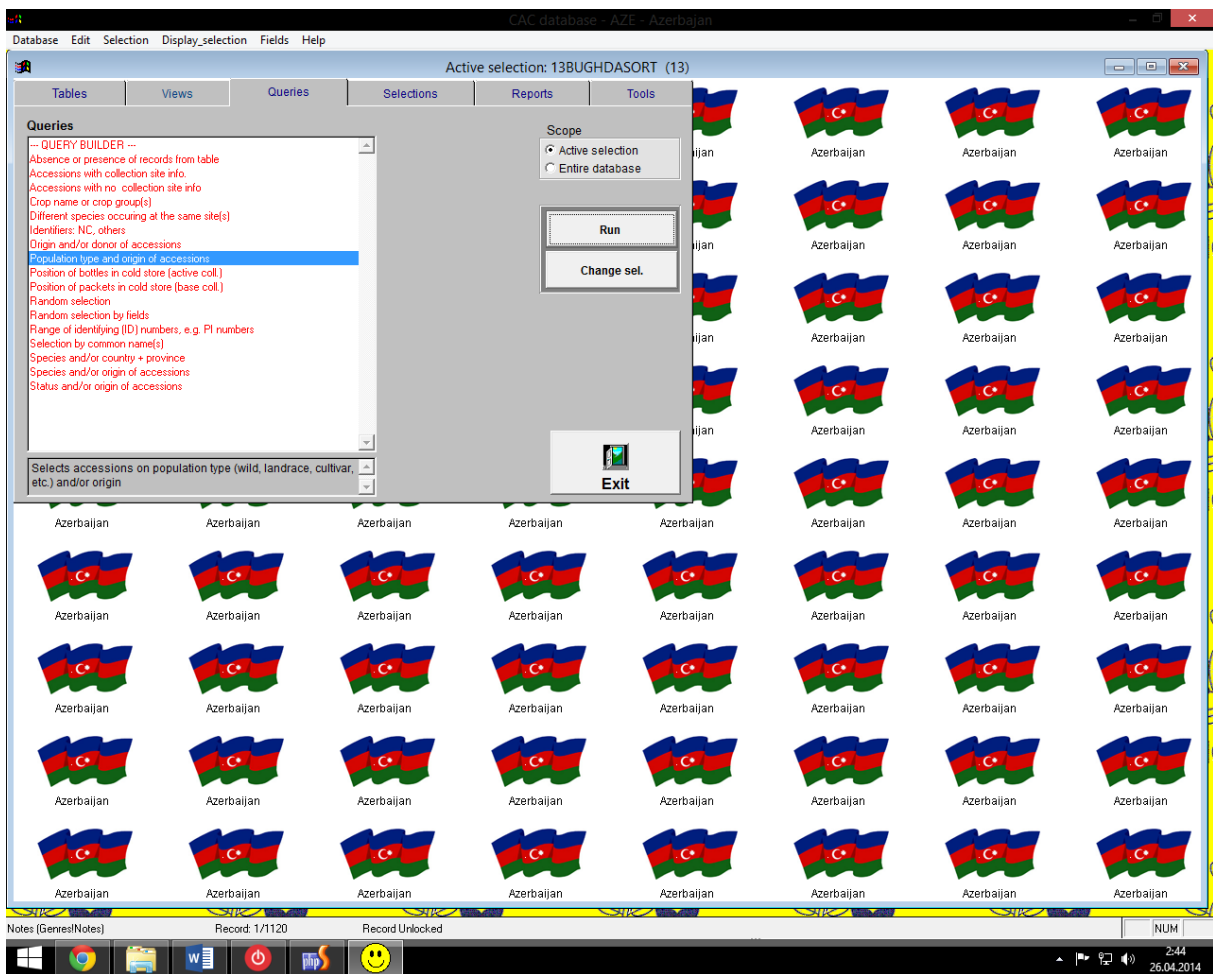
1.2 Nähtavad tulevikuraskused taskukohase arenduse tellimisel

Kuna VFP ei ole enam nii laialdases kasutuses, kui 20 aastat tagasi, tegeletakse selle arendusega tunduvalt vähem, kui alternatiivide puhul. Sellest tulenevalt on VFP platvorm paremal juhul majanduslik risk ja ebamugavus, halvemal mõlemad eelnimetatud ja lisaks veel arvestatav turvarisk. Sellegi poolest ei ole kõik veel kadunud ja arendajaid, kes VFP rakendusi ka pärast 2015. aasta jaanuarit toetama on nõus, leidub. (Basis International, 2014; HostingXChange, 2012) On ka firmasid, mis pakuvad migratsiooniteenust. (EPS Software Corp, 2014) Samas hoiatavad need samad firmad oma kodulehtedel, olgugi, et neil võiks olla majanduslik huvi VFP elu pikendada *ad infinitum*, et kuigi VFP ei ole üldse kehv tarkvara, on tänasel päeval uut VFP arendust alustada perspektiivitu. (Dochke, 2012) Võib-olla ehk anekdootliku, kuid siiski piisavalt hirmutava ettekuulutuse tulevastest arendusraskustest saab CV Keskuse kaudu otsinguid läbi viies. Märksõna „FoxPro“ leiab küll palju vasteid, kuid väga vähesed neist on arendajate CV'd. Aktiivseid töötajaid oli 30.03.2014 seisuga vaid 1 ja temagi näol oli tegemist müügiinimesega ja kuigi esimese 100 tulemuse hulgas oli kaks 19-aastast, on 100 esimese ükskõik millise FoxPro kogemusega inimese keskmine vanus 35,65 aastat. (CV-de otsing märksõnaga "FoxPro", 2014) Võrdluseks: viies läbi samasuguse otsingu ka juba väärikas eas, kuid elujõulise tarkvara kohta, kasutades terminit „MySQL“, ei leidnud autor küll paraku ühtegi aktiivset töötajat, kuid ometi pea ainult sobivaid kandidaate, kellest 100 esimese vaste keskmine vanus oli 27,97 aastat. (CV-de otsing märksõnaga "MySQL", 2014) Autori julgeb endagi, olgugi et lühikesele 9-kuulisele kogemusele tarkvaraarendajana toetudes kinnitada, et odav ei saaks VFP arenduse tellimine juba praegu olema ja tulevikus läheks see aina kallimaks. Lisaks võib juhtuda, et sobiva arenduspartneri leidmine, eriti pikemateks arendustsükliteks või teisemal sügis-talvisel perioodil osutub sootuks võimatuks. Pole kahtlust, et rakenduse portimine mõnele kaasaegsemale levinumale vabavaralisele platvormile oleks majanduslikult hea valik. Seda enam, et töö on jõukohane ühele tudengile mõningate kuude jooksul.

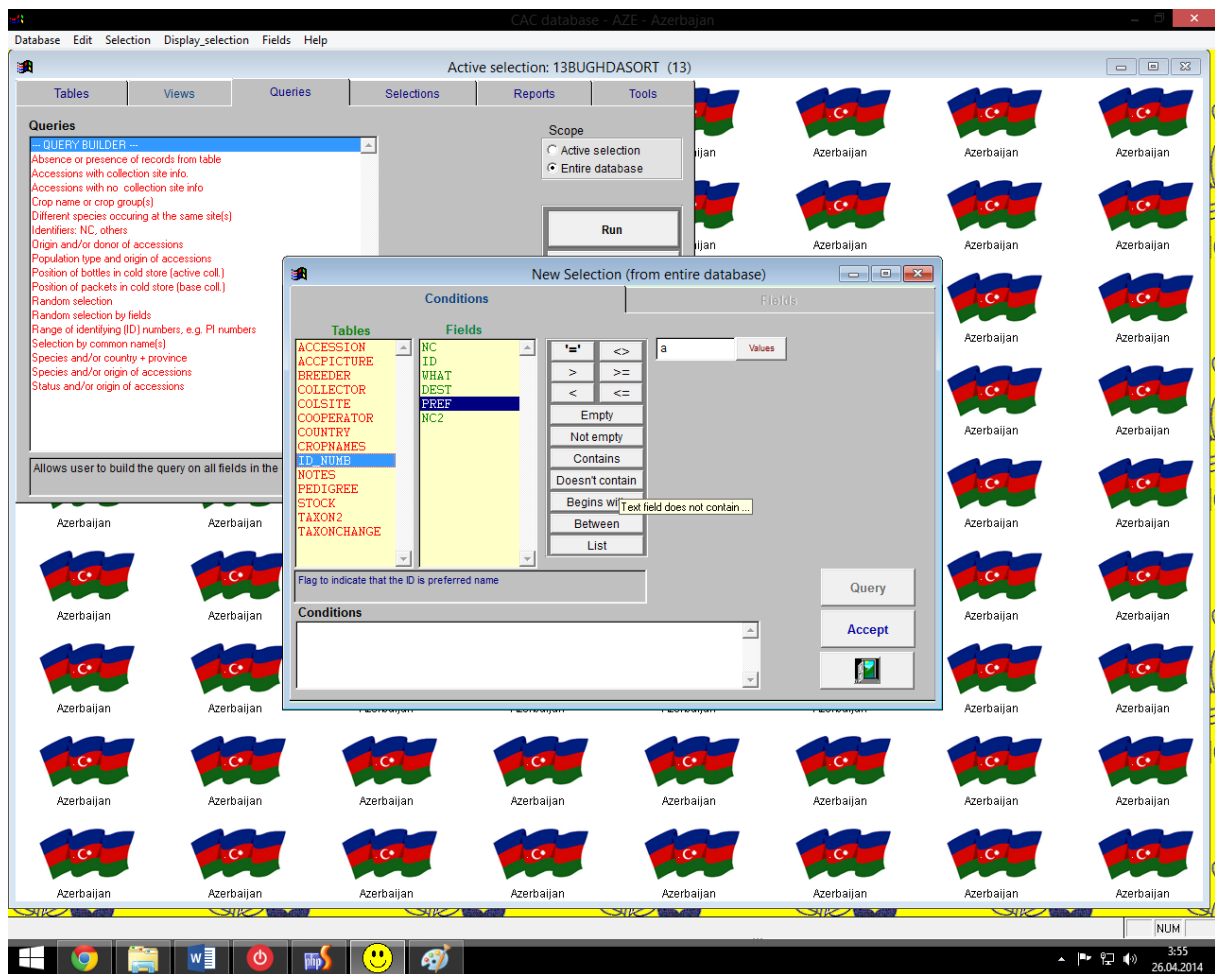
1.3 Windows'i töölaarakenduse eripärad

Olemasoleval rakendusel ei olnud veebileidest, mis raskendas andmetele ligipääsu ja muutis võimatuks tsentraliseeritud „ametlike“ andmete hoidmise. Kasutajad pidid töölaarakenduse endale hankima ja kõigil, kellel see olemas oli, olid ka tõenäoliselt erinevad andmed, mida nad ise aegamööda kas korrigeerisid või siis mitte. Rakendus töötab vaid Windows'i operatsioonisüsteemil ja olgugi, et vähemalt autorile avaldas Windows sel kohal oma tagasiühilduvusega muljet – nimelt käivitus rakendus probleemideta operatsioonisüsteemis Windows 8.1, ei ole Windows tänasel päeval kaugeltki ainus operatsioonisüsteem.

1.4 Võrdlemisi keeruline kasutajaliides ja paigaldusprotsess



Joonis 1: Pärandrakenduse kasutajaliides



Joonis 2: Päringu koostamine VFP rakenduses

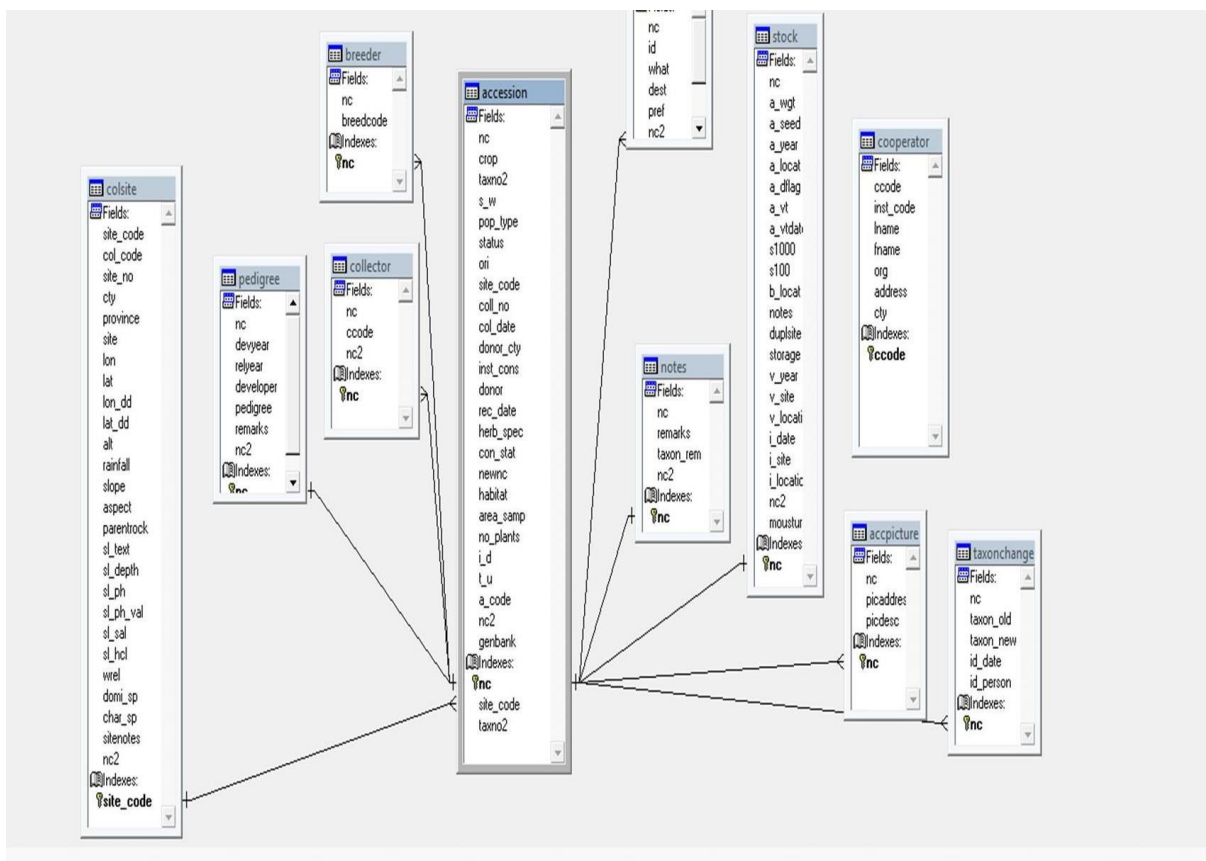
Suhteliselt kirju kasutajaliides sisaldab sadu väga detailseid valikuid. (Joonis 1; Joonis 2) Paigaldusprotsess on järgmine:

- 1) Laadida alla rakenduse failid (392 MB 2394 failina)
- 2) Paigaldada failid C:/CAC kataloogi
- 3) Liigitada C:/CAC/dlls kataloogi sisu C:/Windows/System32 kataloogi
- 4) Käivitada rakendus

Sammude läbimine võtab aega kuni pool tundi. Uue, arendatava rakenduse kasutamine eeldab loodetavasti vaid üksikõik millise tänapäevase brauseri olemasolu, mis toetab JavaScript'i.

1.5 Halvas seisukorras andmebaas

Veel üks probleem oli andmete mittevastavus standarditele, näiteks leitud andmebaasis kümneid erinevaid kuupäevaformaate, mõned neist ei vastanud ühelegi konventsioonile või olid sõna otseses mõttes korrumppeerunud andmed à la „2011/ / „või „ / / /“. Sellised vead muutsid võimatuks näiteks andmete korrektse sorteerimise kuupäeva järgi. Lisaks olid kõik andmed string tüüpi, kaasaarvatud identifikaatorid, kuupäevad ja numbrilised suurused. Andmebaas oli normaliseerimata ja sisaldas suures koguses duplikaatandmeid. Kuigi andmete korrastamisprotsess töötab kesta veel kuid, kui mitte aastaid, on autor suutnud juba viia andmemahu 5332 kilobaidilt 2439-le, kaotamata selle käigus informatsiooni. Andmebaasis puudus mõistlik jaotus olemiteks, paljud neist olid tabelitesse kokku liidetud. See küll aitab vähendada andmebaasipäringute keerukust, kuid pärsib pea kõike muud jõudlusest haldamise lihtsuseni. Suuresti sellest tuleneski vajadus hoida duplikaatandmeid. Näiteks ühe hoidlas oleva seemnesordi liiginime vahetamiseks tuli käia üle kõik selle kohta käivad andmebaasiread ja seal see nimetus ära muuta. Mõnes tabelis esines sisuliselt mitu primaarvõtit, lisaks olid väljade nimed pandud enamasti lühidust silmas pidades ja asjasse pühendamatul oli võimatu saada tabelile peale vaadates aimu, mis andmeid seal kajastatakse. (Joonis 3)



Joonis 3: Andmebaasi struktuur VFP rakenduses

2. Arenduse ettevalmistus ja planeerimine

2.1 Andmete eksport VFP rakendusest

Kuna selgus, et VFP rakenduse edasiarendus ei ole mõistlik, otsustasime Ilhama Mirzaliyeva'ga pärandrakendusest andmed teisaldada. Õnneks võimaldab VFP eksportida andmeid paljudesse erinevatesse vormingutesse ja andmete MySQL andmebaasi kandmine käis mugavalt ning kähku. Esmalt kopeeriti andmed käsuga „EXPORT TO *FAILI_NIMI* TYPE XLS“ Excel'i tabelisse ning sealt edasi .sql tömmisfaili. Tömmisfail sai edasise arenduse sisendiks.

2.2 Nõuete väljaselgitamine

Arenduse tellijal, Ilhama Mirzaliyeva'l, oli väga selge ettekujutus tulevasest rakendusest ja autorile oli intervjuust temaga suurt abi. Sooviti, et veebirakenduses säiliks vana rakenduse funktsionaalsus mõningate lisadega nagu näiteks tõlked, mugav haldusmoodul ning andmete import ja eksport levinud tänapäevastes vormingutes. Väga hea näitena mainis intervjuueeritav EURISCO otsingut lehel http://eurisco.ecpgr.org/search/advanced_search.html. Kuna tellija on ka ise tulevane kasutaja, olid tema eelistused rakenduse disaini suhtes headeks lähtepunktideks.

3. Nõuded

Nõuete väljaselgitamine toimus 2013. aasta detsembris. Autor viis läbi mitu intervjuud Ilhama Mirzaliyeva'ga, uuris hetkel veebis kättesaadavaid analoogseid lahendusi aadressidel <http://barleyregister.icarda.net/public/>, http://eurisco.ecpgr.org/nc/search/advanced_search.html ja ka vana rakendust. Autor püüdis nõuded reastada tähtsuse järjekorras.

1. Kättesaadavus
 - 1.1. Andmed on veebis kättesaadavad.
2. Lihtotsing
 - 2.1. Kasutaja saab viia läbi lihtsaid otsinguid võtmesõna alusel.
 - 2.2. Kasutaja saab eksportida otsingutulemusi Excel'i failidena.
 - 2.3. Kasutaja saab otsingutulemusi sorteerida.
3. Detailotsing
 - 3.1. Kasutaja saab koostada keerulisi päringuid mitme parameetri alusel, rakendus on sisuliselt päringukoostur andmebaasile.
 - 3.2. Kasutaja saab eksportida andmeid ka muudes formaatides.
4. Tõlked
 - 4.1. Rakenduse vaateid on võimalik tõlkida aserbaidžaani ja inglise keelde.
 - 4.2. Andmebaasiväljad on tõlgitavad.
5. Haldus
 - 5.1. Andmebaasitabelid on hõlpsalt hallatavad.
 - 5.2. Andmeid saab .xls või .csv failidena massiliselt importida.
6. Seadistused
 - 6.1. Rakendus on lihtsasti seadistatav, vajadus koodi ümberkirjutamiseks on väike.
7. Turvalisus
 - 7.1. Süsteemi saab turvaliselt sisse logida.
 - 7.2. Rakendusel on administreerimiseks vajalik lehestik.
 - 7.3. Andmeid saavad hallata ja lisada vaid vastavate kasutajaõigustega inimesed.
8. Paindlikkus
 - 8.1. Rakendus on nii dünaamiline, et tabelite ehitust, näiteks väljade nimesid, saab muuta koodi puutumata.

9. Statistika

9.1. Rakendus võimaldab teha statistilist analüüsi, näiteks koostada sagedustabeleid mingite parameetrite alusel.

9.2. Rakendus genereerib statistika põhjal jooniseid ja skeeme.

9.3. Rakendus kogub kasutusstatistikat, näiteks läbi Google Analytics'i.

10. Geoinfo

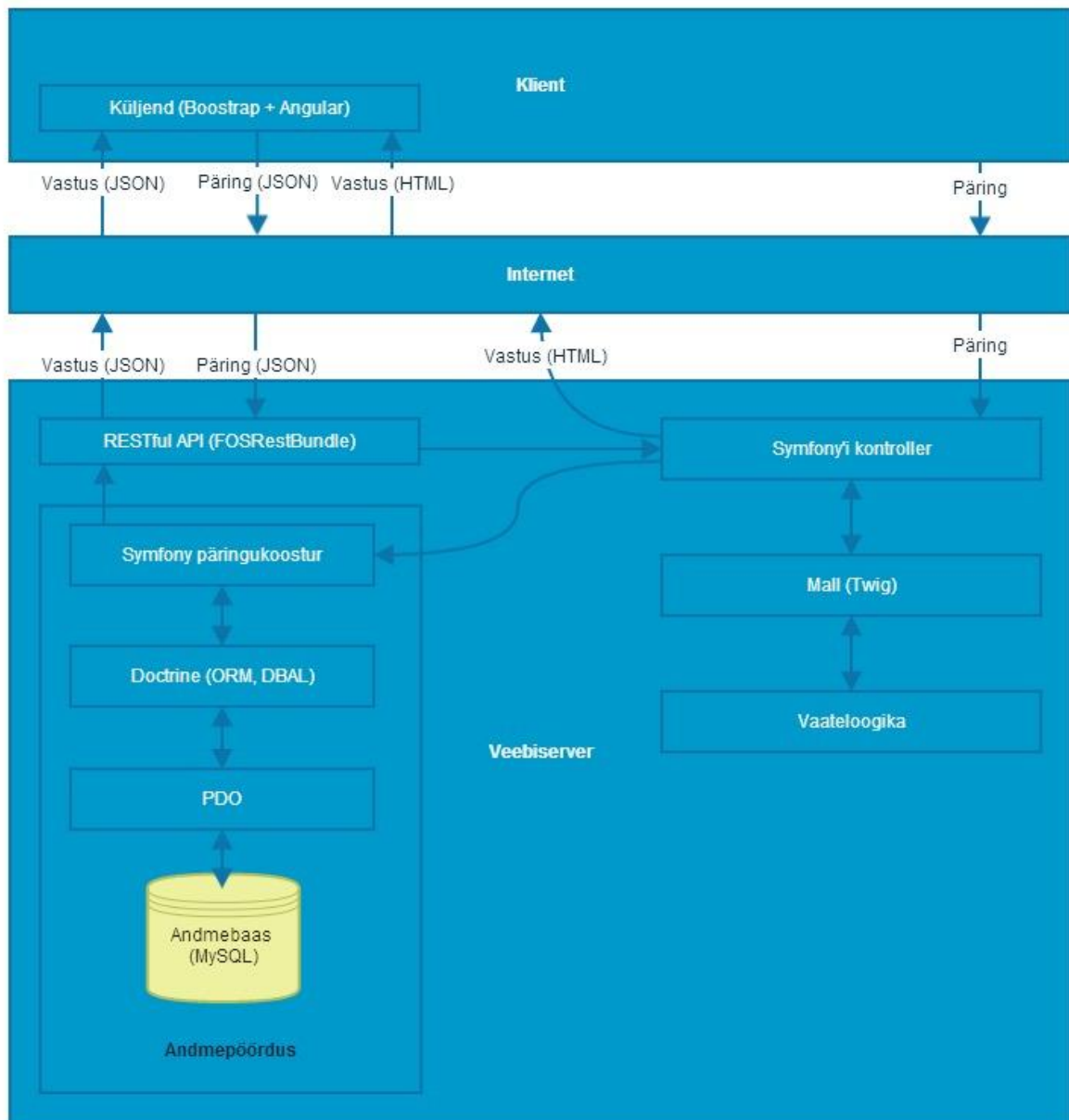
10.1. Kui vajalik informatsioon on andmebaasis olemas, kuvada geneetilise materjali leiukohti interaktiivsel kaardil.

11. Puhverdamine ja *offline* töö

11.1. Otsingud puhverdatakse, et vältida serveri koormamist samade parameetritega otsingutega.

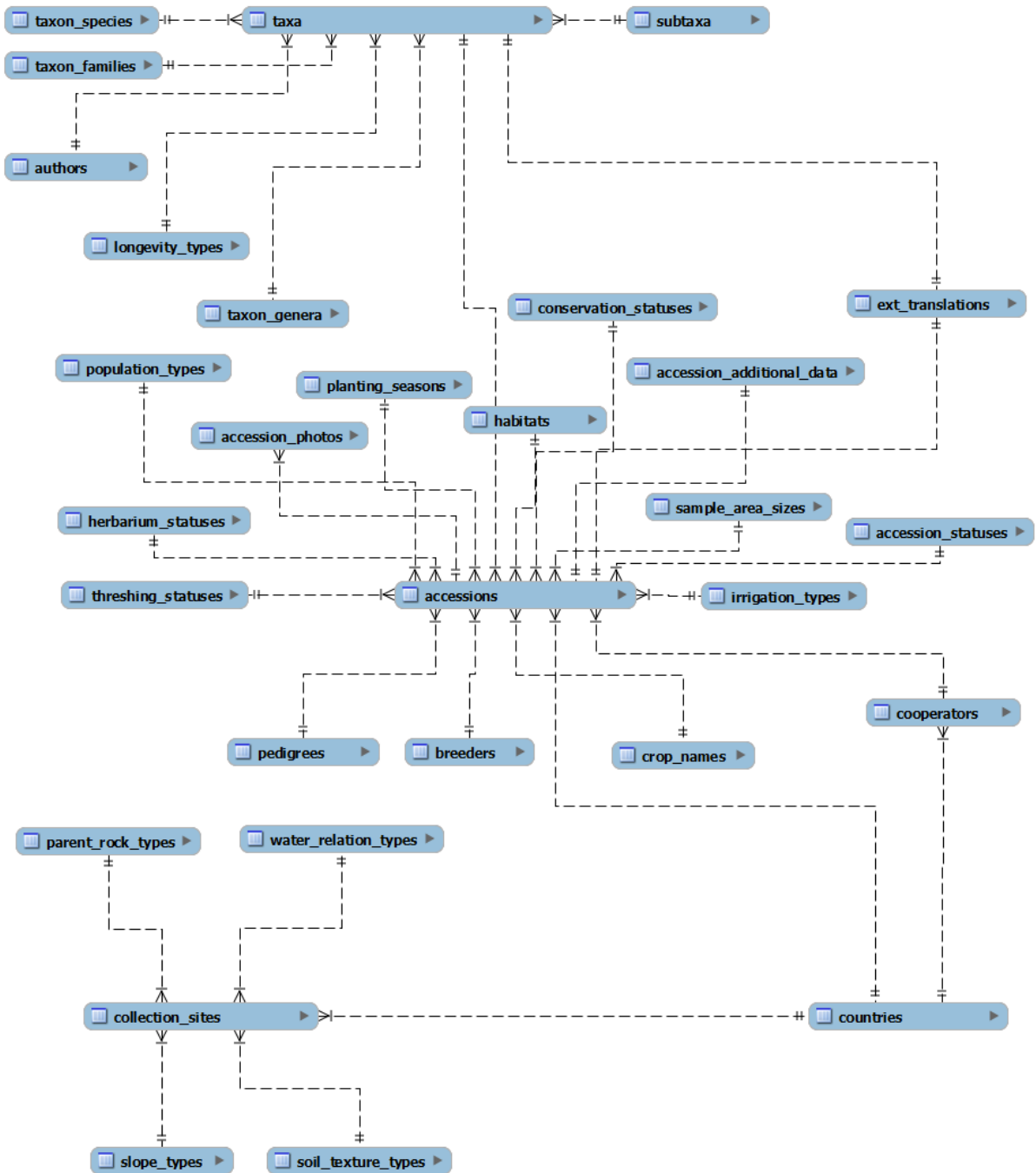
11.2. Võimalik on puhverdada terve andmebaas näiteks HTML5 Web Storage'i abil, nii, et rakenduse põhifunktsioone saab kasutada ka ilma internetiühendusega, näiteks välitöödel, kus traadita interneti levi on kehv.

4. Arhitektuur



Joonis 4: Rakenduse üldine arhitektuur

Ülaltoodud lihtsal arhitektuuriskeemil (Joonis 4) võiks kõige huvipakkavam osa olla JSON suhtlus küljendi ja andmebaasi vahel. Andmeid laetakse kord visualiseeritud mallile asünkroonselt, JSON vormingusse serialiseeritud objektide näol. See vähendab rakenduse komponentide vahelisi sõltuvusi, säästab vähesel määral ka ressursse ja jätab kasutajale lehest oluliselt kiirema mulje (tervet lehte ei värskendata).



Joonis 5: Uus andmebaasi olemi-suhte diagramm

Andmebaasi normaliseeritud struktuur on küll pealtnäha keeruline (Joonis 5), kuid tegelikult arendajale oluliselt kergemini hoomatavam, kui vana skeem. Peamised tabelid on „accessions“, „taxa“ ja „collection_sites“, mis sisaldavad vastavalt informatsiooni geneetiliste varade, taksonoomia ja leiualade kohta. Nende tabelitega on seotud valdavalt üks-mitmele seoses erinevaid väljasid, mida oli mõistlik eraldada oma tabelitesse, et neid oleks parem hallata ning tõlkida. Lisaks vähenes oluliselt andmebaasi suurus. „ext_translations“ sisaldab üks-ühele seoses tõlkeid ükskõik millisele olemile, kuid on töö kirjutamise hetkel kasutusel vaid hoidiste ja taksonite tõlgete tarbeks. Rohkemate seostega tabel on veel „countries“ – riigiga võib olla seotud hoidis ise (hoidise enda asukoht), leiukoht (kust hoidis korjati) või koostööd tegev instituut. Olemi-suhte diagramm sisaldub töö lisades (Lisa 1) ja avaneb tarkvara MySQL Workbench abil. Kui selle tarkvara kasutamine peaks mingil põhjusel võimatu olema, leidub lisades ka andmebaasi tõmmisfail. (Lisa 2)

5. Arendusvahendite valik

Autoril olid vabad käed rakenduse tehnilise teostuse osas ja suuresti said valikud tehtud tema isikliku kogemuse põhjal, kuid olulisteks kaalukeelteks olid ka vahendite veakindlus, pikaajalise toe tagatus ja komponentide laiendatavus. Lisaks pidas autor oluliseks eesmärgiks teha andmed kättesaadavaks võimalikult mitmes vormingus ning eraldada rakenduse kuvade kood andmepääsukoodist, et ühte või teist oleks võimalikult lihtne välja vahetada, kui selleks tulevikus vajadus peaks tekkima ja nii vähemalt mingil määral vältida eelnevale rakendusele osaks saanud saatust. Autor püüdis parimal võimalikul määral kasutada ära isiklike töökogemuse ja töö käigus omandatud oskusi ning teadmisi.

5.1 PHP

PHP on ilmselt kõige levinum veebiprogrammeerimise keel, 2013. aasta jaanuaris kasutas PHP'd 2,1 miljonit serverit ja 240 miljonit veebilehte üle maailma. (PHP, 2014) PHP on üks lihtsamaid programmeerimiskeeli ja kui üldse eksisteerib arendajate küllust, siis seda PHP arendajate hulgas. Kuigi paljud peavad PHP'd „halvimaks peavoolu programmeerimiskeeleks“ (Munroe, 2012) tähendab selle laialdane levik, et leida kedagi, kes rakendust edasi arendaks, oleks tulevikus väga lihtne ja kõigi eelduste kohaselt ka üks kõige odavamaid variante. Lisaks ei ole PHP sugugi nii halb keel, et see rakendusele ka väga pikas perspektiivis sobimatuks võiks osutada.

5.2 Symfony 2.3

Symfony on PHP veebiraamistik, mis ühendab endas ennast PHP veebiarenduse maailmas tõestanud komponente nagu Propel või Doctrine, PDO, PHPUnit, Twig, Swift Mailer jpt. Symfony arendamist alustati 2005. aastal (Symfony, 2014) ja näiteks SitePoint'i lugejate seas läbi viidud küsitluse järgi eelistab seda raamistikku 10,62% vastajatest Laravel'i (25,85%) ja Phalcon'i (16,73%) järel. (Skvorc, 2013) Autor valis Symfony suures osas isikliku eelistuse ajel, kuid arvestada tuleb, et arendaja tööaeg on kallis ja uue raamistiku õppimata jätmine võib olla ka oluline majanduslik argument ja mõjutada projekti valmimisaega. Populaarsemate alternatiivide vastu räägivad veel mõned faktid. Phalcon'i võimekus kompileerida PHP koodi C koodiks ja seega muuta rakendusi palju kiiremaks ja vähem ressursinõudlikuks on ahvatlev, kuid Phalcon'i idee sündis alles 2011. aastal ja kasutamiskõlblikkuse saavutas raamistik alles 2013. aastal, nii noorelt tehnoloogialt ei oleks õiglane oodata parimat stabiilsust, kommuuni, mille foorumites on igale küsimusele vastus leitud või täielikku dokumentatsiooni, Symfony puhul on need kõik aga reaalsus. Samu argumente võib ka kasutada Laravel'i vastu, mille esimene redaktsioon lasti välja 2012. aasta veebruaris. Lisaks on Symfony 2.3 raamistiku pikaajalise toega versioon, mida SensioLabs lubab toetada 2016. aasta maikuuni. (Potencier, 2013) Taaskord on oluline märkida, et pikaajalise toega tehnoloogiate valimine aitab vöidelda vajadusega rakendust moderniseerimise eesmärgil ümber programmeerida, lisaks on tõenäosus, et raamistiku arendajate entusiasm lahtub ja vajalikud veaparandused jäävad tegemata või tehakse neid liiga kaua ja harva, võrratult väiksem. Symfony'ist pärinevad muu hulgas arendatava rakenduse

andmeturbe- ja lokaliseerimiskomponendid, marsruutimine, andmepääsuklassid, häälestusfailid, mallide süsteem ja sõltuvuste sisestus (*dependency injection*). Võib öelda, et Symfony moodustab rakenduse selgroo.

5.3 FOSRestBundle

Tegu on Symfony'le lisatavate komponentide komplektiga, mis võimaldab lihtsat ja kiiret RESTful API-liideste arendamist. (FOSRestBundle, 2014) Rakenduses liiguvad andmebaasiolemid objektide kujul läbi FOSRestBundle'i klasside, et neid saaks pruukida läbi RESTful rakendusliidese.

5.4 MySQL

Andmebaasimootoriks valis autor MySQL'i, kuna tal on sellega suurimad kogemused ja autori hinnangul ei oleks mõne alternatiivi kasutama õppimine ennast ära tasunud. Autor pidas tõsiseltvõetavaks alternatiiviks PostgreSQL'i. Autor analüüsis lühidalt ka populaarsest NoSQL voolust välja kasvanud MongoDB kasutuselevõttu, kuid leidis, et kuigi MongoDB'le lisati täistekstindekseerimine versioonis 2.4 (ilmus 17. aprillil 2013), mis on rakenduse nõuetesse kirjutatud, on see veel pisut toores ja autorile mitte nii sobiv kui MySQL'i täistekstiotsing.

5.5 AngularJS

AngularJS on täiesti iseseisev MVC (*model-view-controller*) raamistik, kuid autor otsustas seda kasutada vaid kasutajaliidese ehitamiseks. Raamistikul on sisse ehitatud võimekus suhelda RESTful rakendusliidestega ja tänu kahepoolsele andmete sidumisele on AngularJS'iga arendamine väga kiire. (Lesh, 2014) Töötava rakenduse loomiseks on tarvis väga vähe koodi. (O'Connell, 2014) Rakenduse saaks edukalt lõpuni ehitada ka vaid Symfony enda komponente kasutades (Twig või PHP mallid), kuid autori hinnangul avanes siin võimalus muuta rakenduse ees- ja tagasüsteemid üksteisest sõltumatumaks. Nimelt suhtleb AngularJS ainult RESTful rakendusliidese ja ehitab selle põhjal üles kõik kasutajale nähtava, seega ei ole põhimõtteliselt oluline, kust AngularJS'ile andmeid ette serveeritakse. Rakenduse puhul toimib see FOSRestBundle poolt JSON vormingusse serialiseeritud objektide näol. Samuti ei tea Symfony tagasüsteem midagi sellest, millega tegeleb AngularJS, tagasüsteem võtab vastu vaid serialiseeritud andmeid, jällegi läbi RESTful rakendusliidese. See peaks taaskord kindlustama rakendust muudatuste vastu, sest võimalik on vahetada välja vaid üks osa, jättes teine puutumata. AngularJS'i Javascript'is kirjutatud koodi on ka põhimõtteliselt lihtsam testida ja siluda kui PHP (või nii mõnegi alternatiivi) koodi.

5.6 Bootstrap

Bootstrap on kogum Javascript'i, LESS'i, SASS'i ja CSS'i failidest, mis kätkevad endas paljusid valmis visuaalseid veebikomponente eesmärgiga kiirendada prototüüpimist ja võimaldada mitte-eksperdil kiiresti luua ka mobiilsetele seadmetele sobivaid adaptiivseid kuvasid. (Bootstrap (front end framework), 2014)

5.7 GitHub

Versioonihaldus on tänapäevase arendusprotsessi loomulik osa ja koodi hoidmine GitHub'i keskkonnas on üks lihtsamaid viise kindlustada koodi säilimine, koostöö eri arendajate vahel ja ülevaade projektist kõigile huvilistele. Bakalaureusetöö kirjutamise ajal on rakenduse lähtekood kättesaadav <https://github.com/Lauriy/plant-genetic-resources> aadressilt.

6. Arenduse käik

Rakenduse arendus on olnud pidev iteratiivne protsess, mis algas 2014. aasta jaanuaris. Bakalaureusetöö kirjutamise ajal on projekti GitHub'i repositooriumis 28 kehtestust, esimene kuupäevaga 12. jaanuar, viimane 4. maist. Autor usub, et koodibaasi tuleks käsitleda kui pidevalt arenevat „organismi“ ja seega ei ole mõtet kunagi öelda, et nüüd on kõik valmis. Siinses bakalaureusetöös esitatud koodi võib pidada prototüüplahenduseks. Autoril on plaanis arendust pärast bakalaureusetöö kaitsmisi jätkata. Projekti kood on 81.1% ulatuses PHP, 17% JavaScript ja 1.9% CSS, mille loomisesse on autor panustanud minimaalselt 37 tundi.

Autoril kulub kõige rohkem aega andmebaasi normaliseerimiseks ja korrastamiseks - üle 20 tunni - ja seda tööd ei saa veel lõpetatuks lugeda. Paraku ei olnud võimalik muude komponentide arendusega enne alustada, kui andmemudel paika sai ja seega ei olnud võimalik ka andmebaasi korrastamist fraasiga „see pole arendaja ülesanne“ tõrjuda ning nii-öelda pead liiva alla peita. Autor ei oodanud, et andmete korrastamine nii suure osa töötundidest võib nõuda ja seetõttu nihkusid teiste komponentide valmimistähtajad edasi.

Kuna arenduse tellija ei ole bakalaureusetöö esitamise ajaks veel veebimajutust leidnud, võib prototüüplahenduse leida autori tööandja, Red Function OÜ serverist <http://lauripage.redfunction.ee> aadressil.

Koodimiseks kasutas autor PhpStorm integreeritud programmeerimiskeskonda, tööajaarvestuseks Togg'it, Debian Wheezy serverite halduseks ja seadistamiseks putty't.

7. Testimine

7.1 Testimisprotseduurid

Autor suhtles ja suhtleb edaspidigi töö tellija, Ilhama Mirzaliyeva'ga e-kirjade teel ja iga kord, kui GitHub'i koodi lisati, testisid rakendust nii autor kui tellija. Kuna kogu rakenduse funktsionaalsus oli töö kirjutamise hetkel ühele lehele mahutatud, ei olnud sellise üldlase testimise läbiviimine kuigi keeruline. Avastatud vead on enamasti olnud kergemat sorti, näiteks ei reastatud täistekstotsingu tulemusi võtmesõnale vastavuse järjekorras ja päringukoostur jäi hätta tühjaks jäetud väljade kasutamisega disjunktiivsetes andmebaasipäringutes.

Lisaks kasutas autor automaatseid kvaliteedikontrollitööriistu, näiteks WAVE'i ja Google'i PageSpeed Insights'i, et kontrollida, kas rakendus järgib ligipäasetavuse kõige põhilisemaid nõudeid ja kas rakendust saab kuidagi kiiremini laadima panna. Töö kirjutamise ajal läbis rakendus WAVE'i ligipäasetavuse automaattestid. (WAVE Report, 2014) Mõned hoiatused jäid õhku, kuid need on tingitud autori disainivalikutest ja ei tohiks oluliselt segada näiteks vaegnägijal lehekülje kasutamist. Tulevikus pöörab autor ligipäasetavusele rohkem tähelepanu, nii, et test ei esitaks ka ühtegi hoiatust ning kliendi soovil viib sisse ka reaalse abivahenditega lehe testimise. PageSpeed hindas lehekülje kiiruse suhteliselt heaks, tasemele 80/100 (PageSpeed Insights, 2014), autor oskaks sealt küll veel mõned punktid täita, kuid need ei oleks aktiivse arenduse ajal mõistlikud käigud. Näiteks JavaScript'i failide kompressimine tuleks siis läbi viia pärast igat muutmist. Tõsi, Symfony võimaldab selle protsessi automatiseerimist ja puhvrite taasloomiseks võib kasutada igasuguseid võtteid, kuid siiski ei leia autor, et veel lähiajal saaks süsteemi täielikku valmidusse viia, kuna see muudab arenduse piisaval määral ebamugavaks.

7.2 Nõuetele vastavus

Ei ole saladus, et vastavas peatükis (Nõuded) esitatud nõuded on töö esitamise tähtjaks vaid osaliselt täidetud. Rakenduse kasutaja saab üsna mitut parameetrit kombineerides andmeid filtreerida, sorteerida ja päringute vastuseid eksportida Excel'i failidena. (Joonis 6) Filtreid võiks küll olla rohkem. Rakendusel puuduvad kirjutamise hetkel administreerimisliidesed, muudatusi tuleb teha otse andmebaasis. Ülejäänud nõuete realiseerimine jääb tulevikku. Siiski on tellija senise tööga väga rahul. Arvestades, et rakendus töötab virtuaalmasinas üsna keskpärase arvuti peal, toimuvad otsingud väga kiiresti. Kuna Debian Wheezy ei toeta PHP 5.5 koos APC puhverdamisega, ei puhverdata otsinguid ja lehelaadimisi, veidi üllatuslikult ei mõju see märgatavalt negatiivselt rakenduse jõudlusele. Seda võib seletada andmete suhteline vähesus. Rakenduse kuvad ja ka olemid on tõlgitavad. Rakendus on nii turvaline, kui Symfony võimaldab, autori hinnangul on turvameetmed piisavad ja rakendus ei ole lihtsate võtetega kräkitav.

Plant Genetic Resources

lauripage.redfunction.ee/#

Accessions

Search: Page: Page size:

Filter box

Export to Excel

Listing selections

Textual

Crop name +

Taxon +

Family +

Adoxaceae

Clear

Species +

Genus +

Temporal

Collection date from

2000-05-01

Clear

Collection date to

2014-05-31

Clear

Recording date from

Recording date to

Locational

id name family collectionDate

| id | name | family | collectionDate |
|------|---------------|-----------|----------------|
| 1635 | mooseberry | Adoxaceae | 2003-01-01 |
| 7890 | elder | Adoxaceae | 2006-07-27 |
| 8097 | Wayfaringtree | Adoxaceae | 2006-10-08 |

0:04 5.05.2014

Joonis 6: Uus veebipõhine rakendus

8. Paigaldus

Rakenduse paigaldamine siinse õpetuse järgi peaks olema jõukohane kõigile, kes kunagi on veebiserverit üles seadnud. Kuna näiteks virtuaalmasina loomine ja seadistamine jääb veidi konkreetse rakenduse paigaldamisõpetuse skoobist välja, alustab autor hetkest, mil paigaldaja on käivitanud masina, milles töötab PHP toega Apache veebiserver ja MySQL andmebaas. Õpetuse näited on läbi viidud Debian Wheezy operatsioonisüsteemil.

8.1 Symfony projektikausta loomine

Esmalt on tarvis alla laadida Symfony raamistik, seda võib teha käsitsi, kuid soovitatav on kasutada kausta loomiseks Composer'i abi. Composer'i saab alla laadida, kui masinasse on paigaldatud curl, käsuga:

```
curl -sS https://getcomposer.org/installer | php
```

Et Composer oleks kogu süsteemis kättesaadav, liigutame faili:

```
mv composer.phar /usr/local/bin/composer
```

Olles soovitud kaustas, antud juhul `/home/vagrant/test`, käsime Composer'il alla laadida ja paigaldada Symfony:

```
composer create-project symfony/framework-standard-edition /home/vagrant/test/Symfony 2.3.*
```

Symfony küsib protsessi käigus sätteid nagu andmebaasi draiver, kasutaja ja parool, näites jäeti kõik vaikeväärtustele.

8.2 Projektifailide hankimine GitHub'ist

```
git clone https://github.com/Lauriy/plant-genetic-resources.git foo
rsync -a foo/ ./
rm -rf foo
```

Nende käskudega kloonime repositooriumi, sünkroniseerimine failid ajutisest kataloogist ja kustutame ajutise kataloogi. Kõik failid peaksid nüüd olema omal kohal.

8.3 Serveri juurkausta suunamine Symfony failide pihta

Debian Wheezy operatsioonisüsteemis tuleb selleks luua fail `/etc/apache2/sites-enabled` kausta, antud juhul on sisuks:

```
<VirtualHost *:80>
  DocumentRoot "/home/vagrant/test/Symfony/web"
  ServerName localhost
  ServerAlias localhost
  ServerAlias *
  <Directory /home/vagrant/test/Symfony/web >
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

8.4 Seadistuste kontrollimine

Adressil <http://localhost/config.php> peaks nüüdseks saama kontrollida, kas kõik serveri sätted on Symfony'le sobivad. Esitatud vead tuleb kõrvaldada.

8.5 Lisasõltuvuste paigaldamine

GitHub'ist alla laetud `composer.json` failis on kirjas teegid, mida rakendus vajab töötamiseks, nende alla laadimiseks projekti juurkaustas:

```
composer self-update
composer update
```

8.6 Andmebaasi paigaldamine

Andmebaasi on võimalik luua mudeli põhjal või kasutades siinse töö lisasid (Lisa 1; Lisa 2). Mudeli põhjal käib tabelite loomine nii:

```
php app/console doctrine:schema:update --force
```

Kokkuvõte

Siinse bakalaureusetöö algseks eesmärgiks oli arendada Visual FoxPro pärandrakendusele veebiliides. Lähemal uurimisel selgus, et see ei oleks siiski mõistlik ja otsustati vanast rakendusest andmed teisaldada, et luua uus levinumatel vabavaralistel tehnoloogiatel põhinev rakendus, mis taluks paremini ajahamba puremist ja mille arendamine oleks tulevikus odav ning hõlbus. Autor uuris vana lahendust, teisi sarnaseid lahendusi ja viis kliendiga läbi intervjuud, et välja uurida, milline uus rakendus olema peab. Töös on pikemalt analüüsitud arendusvahendite valikut, kirjeldatud rakenduse planeerimist ja välja toodud arendusel ilmnunud kitsaskohti.

Töö käigus selgus, et andmete korrastamisele läheb tunduvalt rohkem aega, kui autor oskas ette arvata ja seetõttu lükkusid paljud tööd kaugemale tulevikku ehk rakendus vajab veel täiendusi, autor tegeleb nähtavas tulevikus arendusega edasi. Infosüsteem läbis mitmesugused testimised ja töö tellija oli tulemusega äärmiselt rahul.

Autorile pakkus töö huvitavat väljakutset. Märkimisväärseks võib pidada autori otsust kombineerida Symfony't, AngularJS'i, Bootstrap'i, RESTful rakendusliidest ja tavapärast relatsioonilist MySQL andmebaasi. Autori teada ei ole sellist tüüpi lahendus maailmas levinud. Kombinatsioon tõestas ennast veenvalt ja autor julgeb kinnitada, et ei ole ühegi teiste raamistiku või raamistike kombinatsiooniga nii kiiresti töötavat koodi tootnud, ainus ligilähedane kogemus on Django'ga.

Summary

The author found the topic for this bachelor's thesis in Tallinn University's thesis topic registry and took an immediate interest as the work-to-be seemed to combine genetics and computer science and allow the author to create something that would catch more attention than the average questionnaire analysis about the gaming habits of pensioners. The author was also curious about this mysterious ancient technology called „FoxPro“, which, as it turns out, used to be immensely popular in times the author still occasionally mixed up the left shoe with the right.

The original idea was to create a web interface for the legacy FoxPro application, but it turned out to be unfeasible on many levels, as Microsoft is ending support for FoxPro and developers are becoming harder to come by. The decision was made to port the application to a more modern platform. Symfony, AngularJS, Bootstrap and MySQL were chosen as the tools for the job. A not so widely adopted combination as of yet. The author found AngularJS to be an immensely useful tool for quickly building web sites.

This thesis outlines the research, planning and coding done to get the project as far as it is now. Not all the requirements listed were fulfilled, as it took the author an inordinate amount of time to normalize the severely corrupted tables' data, but testing went well and the client was pleased nonetheless. The author plans to continue working on the project in the foreseeable future.

Viited

Basis International. (16. Aprill 2014. a.). *Life after FoxPro?* Allikas: Basis International veebileht: <http://www.basis-europe.eu/en/index.php/life-after-foxpro.html>

Bootstrap (front end framework). (26. Aprill 2014. a.). Allikas: Wikipedia: [http://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](http://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

CV-de otsing märksõnaga "FoxPro". (30. Märts 2014. a.). Allikas: CV Keskus veebileht: http://www.cvkeskus.ee/cvs.php?op=list&cv_filter_hash=cce73bf5c204365ca4294dabac103941

CV-de otsing märksõnaga "MySQL". (30. Märts 2014. a.). Allikas: CV Keskus veebileht: http://www.cvkeskus.ee/cvs.php?op=list&cv_filter_hash=0ca147a7e5ccaed401cccd1840e6a80

Doschke, O. (13. August 2012. a.). *What is Microsoft Visual FoxPro?* Allikas: Experts Exchange veebileht: http://www.experts-exchange.com/Database/Software/FoxPro/A_10670-What-is-Microsoft-Visual-FoxPro.html

EPS Software Corp. (16. Aprill 2014. a.). *End of Life of FoxPro*. Allikas: EPS Software Corp veebileht: <http://www.codemag.com/VFPCConversion/History>

FOSRestBundle. (26. Aprill 2014. a.). Allikas: FOSRestBundle veebileht: <https://github.com/FriendsOfSymfony/FOSRestBundle>

FoxPro. (30. Märts 2014. a.). Allikas: Wikipedia: <http://en.wikipedia.org/wiki/Foxpro>

HostingXChange. (15. Aprill 2012. a.). *How to Extend the Life of your Visual Foxpro Applications*. Allikas: HostingXChange veebileht: <http://hostingxchange.com/how-to-extend-the-life-of-your-visual-foxpro-applications/>

Lesh, B. (30. Aprill 2014. a.). *Embular Part 2 - What's Great About Angular*. Allikas: Try, Catch, Fail: <http://www.benlesh.com/2014/04/embular-part-2-whats-great-about-angular.html>

Meho, L. I. (Jaanuar 2007. a.). The rise and rise of citation analysis. *Physics World*, lk 32-36.

Microsoft. (16. Aprill 2014. a.). *Microsoft Product Lifecycle Search*. Allikas: Microsoft Support veebileht: <http://support.microsoft.com/lifecycle/search/?sort=PN&alpha=foxpro>

Munroe, A. (9. Aprill 2012. a.). *PHP: a fractal of bad design - fuzzy notepad*. Allikas: fuzzy notepad veebileht: <http://me.veekun.com/blog/2012/04/09/php-a-fractal-of-bad-design/>

O'Connell, A. (16. Aprill 2014. a.). *I Coded the Angular Tutorial App in Backbone and it Took 260% More Code*. Allikas: 42Floors: <http://blog.42floors.com/coded-angular-tutorial-app-backbone-took-260-code/>

PageSpeed Insights. (4. Mai 2014. a.). Allikas: Google Developers: <http://developers.google.com/speed/pagespeed/insights/?url=http%3A%2F%2Fflauripage.redfunction.ee%2F&tab=desktop>

PHP. (26. Aprill 2014. a.). Allikas: Wikipedia: <http://en.wikipedia.org/wiki/PHP>

Potencier, F. (3. Juuni 2013. a.). *Symfony 2.3.0, the first LTS, is now available*. Allikas: Symfony veebileht: <http://symfony.com/blog/symfony-2-3-0-the-first-lts-is-now-available>

Skvorc, B. (28. Detsember 2013. a.). *Best PHP Frameworks for 2014*. Allikas: SitePoint veebileht: <http://www.sitepoint.com/best-php-frameworks-2014/>

Symfony. (26. Aprill 2014. a.). Allikas: Wikipedia: <http://en.wikipedia.org/wiki/Symfony>

Vallaste, H. (30. Märts 2014. a.). *e-Teatmik: IT ja sidetehnika seletav sõnaraamat*. Allikas: Heikki Vallaste veebileht: <http://www.vallaste.ee/>

WAVE Report. (4. Mai 2014. a.). Allikas: WAVE Web Accessibility Tool: <http://wave.webaim.org/report#/http%3A%2F%2Fflauripage.redfunction.ee%2F>

Lisad

Kõik alljärgnevad lisad on ka siinse tööga kaasa pandud CD-l.

Lisa 1



PGR.mwb

Lisa 1: Olemi-suhte diagramm (MySQL Workbench)

Lisa 2



pgr.sql

Lisa 2: Andmebaasi tömmisfail

Lisa 3

Lõputöö käigus arendatava tarkvara lähtekood on saadaval aadressil <https://github.com/Lauriy/plant-genetic-resources> GitHub keskkonnas.