

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Karl Olaf Kuldmaa**  
**Rakendusprogrammi loomine „Kasutatavuse  
kogemuspõhise analüüsi meetodile“**  
**Bakalaureusetöö (9 EAP)**

Juhendajad: Kuldar Taveter, PhD  
Anu Piirisild, MSc

Tartu 2024

# **Rakendusprogrammi loomine „Kasutatavuse kogemuspõhise analüüsi meetodile“**

## **Lühikokkuvõte:**

Bakalaureusetöö eesmärgiks oli luua installeeritav rakendus kasutatavuse kogemuspõhise analüüsi meetodi kasutamiseks. Töö tulemusena valmis rakendus, mida on võimalik Windows operatsioonisüsteemiga arvutites paigaldada ja käivitada. Autori tööna valminud rakendus sisaldab tagasüsteemi, andmebaasi ja eessüsteemi. Disaini lahenduse ning osa eessüsteemist tegi arendustiimi kaasatud disainer. Bakalaureusetöö kirjalik osa sisaldab ülevaadet rakenduse nõuetest, kasutatud tehnoloogiast, arendusprotsessist, valminud rakendusest, valideerimisest, väljakutsetest ja jätkuarendustest.

## **Võtmesõnad:**

Kasutatavuse kogemuspõhise analüüsi meetod, EbA meetod, Spring, Angular, Lauaarvuti rakendus, SQLite

**CERCS:** P175 Informaatika

# **Development Of An Application Program For The „Experience-based Analysis Method“**

## **Abstract:**

The purpose of the bachelor's thesis was to make an installable application for implementing the experience-based analysis method. As a result of the thesis an application was made. It is possible to set up and launch the application on computers with Windows operating system. The application created by the author has a back-end, database and a front-end. The design solution and a part of the front-end was made by the designer who joined the development team. The written part of the thesis contains an overview of requirements, used technologies, development process, completed application, validation, challenges and future development.

## **Keywords:**

Experience-based analysis method, EbA method, Spring, Angular, Desktop application, SQLite

**CERCS:** P175 Informaatika

## Sisukord

Sissejuhatus	5
1. Mõisted ja terminid	7
2. Nõuded rakendusele	8
2.1 Rakenduse skoop	8
2.2 Funktsionaalsed nõuded	8
2.3 Mittefunktsionaalsed nõuded	9
3. Kasutatud tehnoloogiad	11
4. Arendusprotsess	13
4.1 Arenduse esimene etapp	14
4.2 Arenduse teine etapp	14
4.3 Arenduse kolmas etapp	14
5. Valminud rakendus	16
5.1 Rakenduse ülesehitus	16
5.1.1 Tagasüsteem	16
5.1.2 Eessüsteem	17
5.2 Installeerimis- ja käivitamisprotsess	17
5.3 Disaini lahendus	18
5.3.1 Projekti haldus	18
5.3.2 Projekti töölehe haldamine	19
5.3.3 Rakenduse kasutusjuhend	21
5.3.4 Tutvustuse vaade	22
5.3.5 Päis	22
5.4 Andmemudel	23
6. Valideerimine	25

7. Väljakutsed	27
8. Jätkaarendused	29
9. Kokkuvõte ja järeldused	30
Viidatud kirjandus	31
Lisad	33
I Küsimustik	33
II Teise etapi küsimustik	37
III Andmed testimiseks	39
Litsents	40

## Sissejuhatus

Kasutatavuse kogemuspõhise analüüsi (EbA) (ingl *Experience-based Analysis*) meetodina on loodud eesmärgiga saada digitoodete või e-teenuse arendamise võimalikult varases faasis aimu, kas lahendus kavandatud moel hakkab toimima – seda just lähtuvalt kasutajate käitumise ja harjumuste seisukohast –, ja millised on võimalikud riskid [1, 2]. Kui nõuete valideerimise käigus keskendutakse tavaliselt sellele, et teada saada, kas arendatav toode vastab sidusrühmade vajadusele, siis EbA eesmärk on teada saada, kuivõrd on sidusrühmadel olemas vajalikud teadmised, oskused, harjumused või vahendid, mis on toote edukuseks vajalikud [1]. EbA meetod on eelkõige mõeldud kasutamiseks nõuete analüüsi läbiviijatele [1, 2]. EbA meetodi autor Anu Piirisild on meetodi loomisel ette näinud, et selle juurde kuulub meetodi rakendamist toetav töövahend. Kui on olemas töövahend, mis aitab meetodit rakendada, on meetodit lihtsam praktikutele igapäevasesse kasutusse pakkuda [3]. Anu Piirisild (edaspidi klient) on loonud prototüübi, mis on arvutustabeli formaadis [1], kuid vajadus on kasutajasõbralikuma rakenduse järele.

Töö eesmärgiks on luua EbA meetodi kasutamiseks rakendus, mida on võimalik nõuete analüüsi tegijatel Windows operatsioonisüsteemis installeerida oma arvutisse. Windows osutus valituks oma laialdase populaarsuse ja piisava kasutajatoe pakkumise tõttu. Töö teostamisel võeti aluseks kliendi loodud prototüüp. Töö tegemisel lähtuti kliendi nõuetest ning tagasisidest. Kogu töö dokumenteeriti arvestusega, et rakenduse teise etapi funktsionaalsuste arendamise võtab üle järgmine tiim. Bakalaureusetöö käigus valminud rakendus ei ole lõplik.

Rakendus koosneb andmebaasist, tagasüsteemist (ingl *back-end*) ja eessüsteemist (ingl *front-end*). Rakenduse disaini lahenduse kavandas ja teostas Iris Kreinin (edaspidi disainer), kes vormistas oma töö protsessi ja tulemused bakalaureusetöösse pealkirjaga „Kasutajaliidese disaini loomine „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakendusele“ [4]. Rakenduse eessüsteemist lõi autor põhilise funktsionaalse osa ning disainer visuaalse osa ja osa disaini lahenduse teostamiseks vajaminevast funktsionaalsusest [4]. Rakenduses kuvatavad tekstid ja juhendmaterjalid koostas klient.

Uuritavateks probleemideks on installeeritava rakenduse loomine ning rakenduse kohaldamine disaini lahendusega. Installeeritav rakendus loodi kliendi sisendi ja olemasoleva prototüübi alusel. Kogu rakenduse nõuetest valiti välja ja võeti töösse nõuded, mis on vajalikud rakenduse põhifunktsionaalsuse saavutamiseks.

Bakalaureusetöö jaguneb kaheksaks sisuliseks peatükiks. Esimeses peatükis tuuakse välja töö kirjutamisel kasutatud mõisted ja terminid. Teises peatükis kajastatakse kliendi poolt esitatud nõudeid rakendusele. Kolmandas peatükis kirjutatakse, millised erinevad tehnoloogiad rakenduse ehitamisel kasutusele võeti ja miks. Neljandas peatükis kirjeldatakse rakenduse arendusprotsessi. Viiendas peatükis tutvustatakse valminud rakendust ja selle komponente. Kuuendas peatükis kajastatakse manuaaltestimise tulemusi. Seitsmendas peatükis käsitletakse arenduse käigus tekkinud väljakutseid. Kaheksandas peatükis on info planeeritavate jätkuarenduste kohta.

## 1. Mõisted ja terminid

**Tagasüsteem** – kasutajale nähtamatu töötlev, talletav ja käitlev põhiosa: tagasüsteem, tagakomponent vms (ingl *back-end*) [5].

**Hüpikaken** – element veebilehel, mis ilmub käivitava koha klõpsamisel või ülelibistusel (ingl *pop-up window*) [5].

**Residentne** – põhimälus asuv või põhimälus toimuv (ingl *in-memory*) [5].

**Eessüsteem** – inimkasutajat või kasutavat süsteemi tagaosaga liidestav (ingl *front-end*) [5].

**UI/UX** – kasutajaliides/kasutajakogemuse (ingl *user interface/user experience*) [5].

**HTML** – veebilehtede loomise vahend, vormingusüsteem, mis võimaldab tekstile lisada märgistuskoodide (siltide) abil kirjatüüpi ja -laadi, värvust, graafikat, hüperlinke jm [5].

**SQL** – deklaratiivne interaktsiooni- ja programmeerimiskeel relatsioonandmebaaside halduseks ja kasutamiseks [5].

**MVC** – tarkvara arhitektuurimudel, mis lahutab kasutajaliidese valdkonnateabest [5].

**Kaardistamine** – vastendamise protsess või tulem (ingl *mapping*) [5].

**Pesastamine** – ühetüübiliste struktuuride samatüübilise sisse paigutamine (ingl *nesting*) .

**WCAG** – veebisisu hõlbustuse juhised, näiteks puuetega kasutajatele [5].

**API** – reeglid ja vahendid rakendusprogrammi suhtluseks [5].

## **2. Nõuded rakendusele**

Rakenduse arenduse tiimi kuulusid kolm inimest: autor, disainer ja klient. Klient täitis ka tooteomaniku rolli. Antud peatükis on välja toodud tööde skoop ja EbA meetodi installeeritava rakenduse funktsionaalsed ja mittefunktsionaalsed nõuded. Kõik rakenduse nõuded kirjeldas klient. Nõuded kirjeldasid rakenduse valmislahendust. Kogu rakenduse nõuetest valiti välja ja võeti töösse nõuded, mis on vajalikud rakenduse põhifunktsionaalsuse saavutamiseks. Nõuded täienesid osaliselt UI/UX disaini analüüsi käigus, mis toimusid ennekõike disaineri ja kliendi eestvedamisel. Peatükis on välja toodud nõuded, mis täideti autori poolt.

### **2.1 Rakenduse skoop**

Käesoleva töö skoobiks määrati luua keskkond, kus rakendus:

1. on allalaetav ning installeeritav Windows operatsioonisüsteemiga arvutites;
2. säilitab andmeid lõppkasutaja masinas;
3. ei vaja oma töö käigus internetiühendust.

Kliendi antud nõuetest valiti arenduse jaoks prioriteetsemad nõuded, mis olid vajalikud EbA meetodi põhifunktsionaalsuse rakendamiseks. Täideti ka nõudeid, mis ei olnud EbA meetodi rakendamiseks kriitilised, kuid kasutatavuse tõstmise jaoks väga olulised. Rakenduse skoopi valitud ja teostatud funktsionaalsused on kirjeldatud alapeatükis 2.2.

### **2.2 Funktsionaalsed nõuded**

Funktsionaalsed nõuded on koostatud kahele peamisele kasutajagrupile: analüüsi läbiviija, kes on rakenduse lõppkasutaja ja kelle eesmärk on teha loodava toote, e-teenuse või projekti analüüsi EbA meetodiga ning tootejuht, kes peab saama rakenduses staatiliselt kuvatavaid tekste täiendada. Funktsionaalsed nõuded on kirjeldatud kasutuslugudena:

Üldised nõuded rakendusele:

1. Analüüsi läbiviijana saan hallata EbA meetodi rakendamise töölehti projektide kaupa, et jagada analüüs loogilisteks osadeks.
2. Tooteomanikuna saan rakenduses kuvatavate staatiliste tekstide tõlkefailide sisu muuta, et parandada kirjavigu või teha tekstis muudatusi.



Nõuded rakenduse töölehe vaatele:

1. Analüüsi läbiviijana saan hallata sidusrühmade nimekirja, et analüüsi tegemisel ei peaks korduvalt nimetusi uuesti kirjutama.
2. Analüüsi läbiviijana saan hallata funktsionaalsusgruppide nimekirja, et struktureerida analüüsitavaid nõudeid ühe projekti piires.
3. Analüüsi läbiviijana saan seada funktsionaalsusele identifikaatori vabatekstiliselt, et identifikaator kattuks mujal projektdokumentatsioonis kasutatava numeratsiooniga.
4. Analüüsi läbiviijana saan väljal "Sidusrühm" lisada sidusrühma loetelust, et analüüsi tegemise ajal ei peaks korduvat sidusrühma nimetust uuesti kirjutama.
5. Analüüsi läbiviijana saan uusi ridu lisada, et samal töölehel kirjeldada mitut funktsionaalsust.
6. Analüüsi läbiviijana saan "Funktsionaalsuse kirjelduse" alla lisada rida, et näidata mitme eeltingimuse korral nende kuuluvust sama funktsionaalsuse alla.
7. Analüüsi läbiviijana saan "Funktsionaalsuse eeltingimuse" alla lisada rida, millega on võimalik näidata erinevate situatsiooni näidete kuuluvust sama eeltingimuse alla.
8. Analüüsi läbiviijana saan tabeli reas vastata neljale küsimusele, igaüks eraldi tabeli lahtris, nelja vastusevariandiga, et teha analüüsi vastavalt EbA meetodi raamistikule.
9. Analüüsi läbiviijana saan ridu kustutada, et kustutada ebavajalik info rea kaupa korraga.
10. Analüüsi läbiviijana saan vabatekstiliselt täita kõiki lahtreid, millele ei ole eraldi tingimusi seatud, et sisestada projektiga seotud infot.

### **2.3 Mittefunktsionaalsed nõuded**

Rakenduse mittefunktsionaalsed nõuded on struktureeritud üldisteks nõueteks ja nõueteks, mis antud töölehe vaatele.

Üldised nõuded rakendusele:

1. Rakenduse kasutada soovija saab rakendust alla laadida.
2. Rakenduse kasutada soovija saab installeerida rakendust Windows operatsioonisüsteemiga arvutites.
3. Rakendus on eesti ja inglise keelne.
4. Rakenduse kasutamiseks ei ole vaja internetiühendust.
5. Rakendusel on juhendi leht.

6. Rakendusel on täiendava info leht.
7. Rakendusega tuleb kaasa näidisprojekt eesti ja inglise keeles

Nõuded rakenduse töölehe vaatele:

1. Esimesed kaks tulp jäävad paigale ning käivad horisontaalselt kerides lehe vasakus servas kaasa.
2. Uue funktsionaalsuse grupi avamisel loob rakendus automaatselt selle alla vaikimisi ühe tühja rea.
3. Esimeses tulbas kuvatakse rakenduse poolt vaikimisi sisestatud rea järjekorranumbrit.
4. Tabeli tulba “Kas” väljad peavad olema rakenduse poolt automaatselt täidetud.
5. Neljale küsimusele vastates kuvab rakendus tulbas “Kirjeldus, mil määral on eeltingimus täidetud” kolmiktulbast esimese tulba alla küsimuste tulemuse kombinatsioonile vastava teksti<sup>1</sup>.
6. Sidusrühma tulba sisu on dubleeritud rakenduse poolt “Kirjeldus, mil määral on eeltingimus täidetud” kolmiktulbast teise tulba alla.
7. Funktsionaalsuse eeltingimuse tulba sisu on dubleeritud rakenduse poolt “Kirjeldus, mil määral on eeltingimus täidetud” kolmiktulbast kolmanda tulba alla.
8. Kombinatsioonide vastuse tekst, sidusrühm ja eeltingimuse tekstid kuvatakse rakenduse poolt “Kirjeldus, mil määral on eeltingimus täidetud” kolmiktulba järel asuvasse veergu kokku kirjutatud lausena.
9. Kui alustada kustutamisprotsessi tulpadest “Funktsionaalsuse kirjeldus” või “Funktsionaalsuse eeltingimus”, millel on alamread, siis kustutab rakendus ka alamread.
10. Rakendus avab abitekstid hüpikakendega (ingl *pop-up windows*).

---

<sup>1</sup> Vastuste kombinatsioonide variandid ja nende sõnastuse koostas klient [1].

### 3. Kasutatud tehnoloogiad

Käesolevas peatükis antakse ülevaade tehnoloogiatest, mida autor kasutas rakenduse loomisel, tehnoloogia valiku põhjustest ning on ära nimetatud, milline osa rakendusest on vastava tehnoloogiaga tehtud. Autor valis ees- ja tagasüsteemi tehnoloogiad arvestades isiklikku kogemust programmeerijana. Autor valis võimalikult palju tehnoloogiaid, mis olid rakenduse jaoks sobilikud ja millega oleks võimalik kõik nõuded katta. Oluliseks faktoriks oli ka see, et tehnoloogial oleks aktiivne hooldus. Aktiivne hooldus on oluline, sest see tagab, et tehnoloogia käib uuendustega kaasas ning parandab tekkinud probleeme. Autoril ei tekkinud valitud tehnoloogiate kasutamisega arenduse käigus probleeme.

SQLite on C programmeerimiskeeles teek, mis teostab väiksemahulist, kiiret, iseseisvat ja kõrge töökindlusega täiemahulist relatsioonilist SQL andmebaasimootorit [6]. SQLite on rakenduses kasutusel andmete hoiustamiseks.

Andmebaasi valikust langesid välja teised andmebaasimootorid nagu PostgreSQL ja MySQL, sest installeeritava rakenduse nõuetest tulenevalt on arendustehniliselt ajasäästlikum kasutada residentset (ingl *in-memory*) andmebaasi. Residentset andmebaasi on vaja, kuna andmebaas jookseb rakenduses sees ning andmebaas eraldi jooksva rakendusena kulutaks ebavajalikult tarbija arvuti ressursi. Eraldi jooksva rakendusena andmebaas nõuab ka lõppkasutaja poolt seadistamist, mida residentne andmebaas ei nõua.

Java on abstraktse tasemega, automatiseeritud mäluhaldusvõimekusega, objektorienteeritud ja üldotstarbeline programmeerimiskeel, mis on mõeldud koodi käitamiseks igas arvutis [7]. Java keeles on kirjutatud rakenduse tagasüsteemi kood.

Spring raamistik pakub laiaulatuslikku programmeerimis- ja konfigureerimismudelit tänapäevastele Java rakendustele kõikides evitamiskeskkondades [8]. Spring raamistikul on loodud rakenduse tagasüsteem.

Maven on tarkvaraprojekti haldamise tööriist, mis on loodud projekti objekti mudeli kontseptsiooni alusel, millega on võimalik hallata projekti ehitamist, raporteerimist ja dokumenteerimist keskest infoallikast [9]. Mavenit kasutati rakenduse kompileerimiseks, sõltuvuste haldamiseks ja pakendamiseks.

TypeScript on tugevalt tüübitud programmeerimiskeel, mis ehitub JavaScript-ile ning pakub staatilisi annotatsioone ja valikulisi tüübiannotatsioone [10]. TypeScript keeles on loodud rakenduse eessüsteem.

Angular on raamistik ja platvorm, millele saab ehitada üheleheküljelisi kasutajaliidese rakendusi kasutades HTML-i ja TypeScript-i. Raamistikus on teostatud põhi- ja valikulised funktsionaalsused imporditavate teekidena [11]. Angulari raamistikule on tehtud rakenduse eessüsteem.

Launch4j on platvormiülene tööriist Java rakenduste pakendamiseks Windows operatsioonisüsteemile sobiva faililaiendiga [12]. Launch4j kasutatakse Windows operatsioonisüsteemiga arvutile käitatava EbA rakenduse loomiseks. Valik tehti mitmeid erinevaid variante proovides kuniks sobiva tulemuse leidmiseni. Valik tehti ka kasutuslihtsuse ja populaarsuse alusel. Launch4j populaarsust näitab see, et rakendust on alla laetud üle 2 miljoni korra [13].

Inno Setup on vabavaraline installeerija Windows operatsioonisüsteemi rakendustele [14]. Inno Setup-it kasutatakse Java keskkonna ning Launch4j-ga loodud rakenduse pakendamiseks installeerijaks. Valik tehti mitmeid erinevaid variante proovides kuniks sobiva tulemuse leidmiseni. Valik tehti ka kasutuslihtsuse ja populaarsuse alusel. Rakendust on alla laetud miljon korda [15].

## 4. Arendusprotsess

Käesolevas peatükis kirjeldatakse rakenduse arendamise protsessi. Arendusprotsess jaotus loogiliselt kolme suuremasse etappi: nõuete analüüs ja algse ees- ja tagasüsteemi ehitamine, ees- ja tagasüsteemi täiendamine rakenduse skooopi valitud funktsionaalsuste saavutamiseks ning täiendavate funktsionaalsuste loomine, testimine ja kriitilisemate muudatuste sisseviimine.

Arenduse keskkonnaks oli autori koduarvuti. Programmeerimiseks kasutas autor Intellij IDEA Ultima kasutajaliidest. Rakenduse kasutamiseks loodud installeerija kasutamist katsetati autori poolt erinevates Windows operatsioonisüsteemiga arvutites.

Rakenduse koodi halduseks kasutati Git-i. Rakenduse koodihoidla on avalikult nähtav ning asub Github keskkonnas<sup>2</sup>. Koodihoidlas on täpselt kirjeldatud autori, kliendi ja disaineri panus koodi kirjutamisel. Koodihoidlas on kirjeldatud rakenduse ehitamise juhend ning juhend kliendile staatiliste tekstide muutmiseks. Arenduseks valiti tüve meetod, et tiimis saaks töötada mitu inimest samaaegselt. Tüve meetod on allika juhtimise mudel, kus arendajad panustavad ühte harusse, mida kutsutakse tüveks. Tüve meetodit rakendades ei teki mitme arendaja samaaegsel tegevusel koodi mestimisel konflikte [19]. Tekkinud tehnilistele küsimustele ja probleemidele leidis autor lahendused erialastest foorumitest, näiteks Stack Overflow-st.

Arenduses tehti funktsionaalsuse realiseerimisel kõigepealt muudatused andmemudelis, seejärel tagasüsteemi rakenduses ning kõige lõpuks eessüsteemi rakenduses. Arenduse alguses tehti teadlik valik jätta erinevad automaattestid tegemata, sest autori eelneva programmeerimise kogemuse alusel pikendab testide kirjutamine arenduse aega vähemalt kolm korda, mille tõttu ei oleks rakendus valminud ettenähtud ajaks. Automaattestide sooritamine on ära mainitud jätkutegevuste loetelus.

Rakenduse arendustiimi sees toimis aktiivne suhtlus autori, kliendi ja disaineri vahel, mille käigus teavitati valminud funktsionaalsustest, jagati ülesandeid ning vahetati omavahel muud rakenduse arendamisega seotud informatsiooni. Kliendile vahetulemuste esitamiseks compileeriti installeeritav rakendus.

---

<sup>2</sup> <https://github.com/karl1245/EUIM>

#### **4.1 Arenduse esimene etapp**

Esimeses etapis analüüsis autor rakenduse nõudeid, valis välja vajalikud tehnoloogiad ja lõi algse rakenduse. Kliendi prototüübist lähtudes loodi andmemudel ning salvestati andmebaasi meetodi kuvamiseks vajalikud lähteandmed.

Ehitati valmis tagasüsteem, mille kaudu sai eelpool mainitud lähteandmeid pärida. Valmis ka eessüsteem, kus oli võimalik projekte hallata ja EbA meetodit rakendada.

Esialgne rakendus oli täiesti dünaamiline, mis tähendab seda, et rakendusele oli võimalik anda ükskõik milline andmemudelile vastav lähteandmestik ja rakendus võttis lähteandmestiku vastu kavandatud viisil.

Esimese etapi lõpuks oli valmis installeeritav rakendus, mille abil oli võimalik eessüsteemi, tagasüsteemi ja andmebaasi ühe rakendusena installeerida. Rakenduse komponentide vaheline suhtlus toimus.

#### **4.2 Arenduse teine etapp**

Teises etapis liitus rakenduse arendusega disainer. Disaineri esimese prototüübi valmides tuli autoril juba olemasolevat rakenduse loogikat täiendada ja muuta. Disainer ja autor töötasid sama koodibaasiga ning jooksvalt täiendati ja parandati üksteise tehtud koodi.

Kliendi poolt esitatud nõuded muutusid töö käigus. Muutused tulenesid põhiliselt disainile kohaldamise vajadusest või tehnoloogilistest piirangutest. Paljudele tulpadele rakendus koodis ja andmebaasis eriloogika, mis tähendas seda, et rakenduse kasutajaliidest ehitati ümber seostades omavahel konkreetseid visuaalsed elemendid ja lähteandmed.

Põhifunktsionaalsuse valmimisel oli vaja hakata disaineri poolt loodud disaini ja autori poolt loodud funktsionaalsust ühildama. Disaini ja funktsionaalsuse ühildamine oli arendustehniliselt kõige keerukam osa.

Teise etapi lõpuks olid skoobis nimetatud funktsionaalsused rakendatud ning disaineri poolt loodud disaini lahendus teostatud [4].

#### **4.3 Arenduse kolmas etapp**

Kolmandas etapis toimus lisafunktsionaalsuste arendamine, rakenduse korrastamine testimiseks ja testimise järgselt kriitilisemate muudatuste sisseviimine.

Rakenduse disaini lahendus täienes töö käigus vastavalt disaineri ja kliendi tehtud kasutatavuse analüüsile. WCAG nõuetele vastavuse kontrollimisel avastas disainer puudused, mis parandati disaini lahenduses. Disainilahenduse täiendused teostati rakenduses.

Arenduse kolmandas etapis korrastati rakendust. Korrastamise käigus sisse viidud muudatused puudutasid valdavalt mittefunktsionaalseid nõudeid. Peale korrastamist oli rakendus valmis testimiseks. Testimise protsess ja metoodika on kirjeldatud peatükis 6. Arenduse kolmandas etapis toimus ka tekstide parandamine ja visuaalse poole parendamine.

Testimise tulemuste analüüsimisel prioritseeriti koos disaineri ja kliendiga kõige kriitilisemad muudatusvajadused. Kriitilised muudatusvajadused teostati rakenduses. Ülejäänud vajadused kaardistati jätkuarendustena ja need on nimetatud peatükis 8.

Kolmanda etapi lõpuks oli valmis rakendus, milles olid peatükis 2 loetletud nõuded täidetud.

## 5. Valminud rakendus

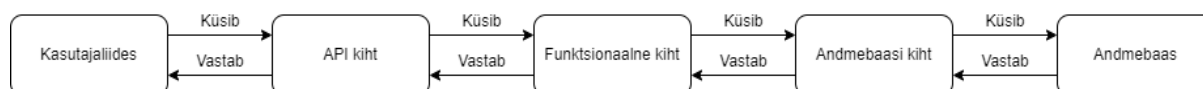
Loodud rakenduse nimeks on EbA-UIM. Peatükk annab ülevaate loodud rakenduse ülesehitusest, komponentidest, installeerimis- ning käivitamisprotsessist, disaini lahendusest ja andmemudelitest.

### 5.1 Rakenduse ülesehitus

Rakendus koosneb tagasüsteemi ja eessüsteemi rakendusest ning andmebaasist. Kõik kolm esitatakse kasutajale ühe rakendusena. Ees- ja tagasüsteemi ühe rakendusena ehitamiseks on kasutatud SpringAngularDemo malli [16]. Koodi dokumentatsiooniks on kood ise, kus erinevad koodi osad, nagu näiteks funktsioonid ja muutujad, on nimetatud selle järgi, milleks neid kasutatakse või mida nad endas sisaldavad. Rakenduses kuvatavaid tekste on tootejuhil või ligipääsuõigusega isikul võimalik muuta koodihoidlas vastavaid tekstifaili muutes või SQL skripte lisades. Rakenduse ülesehitusel arvestati ka potentsiaalsete jätkuarendustega ning võimalusega pakkuda rakendust kesksest serverist.

#### 5.1.1 Tagasüsteem

Tagasüsteemi rakendus põhineb spring raamistikul. Tagasüsteemi rakendus kasutab MVC-l põhinevat ülesehitust. Springi MVC raamistik on disainitud funktsionaalsuse ümber, mis suunab sissetulevad päringud konfigureeritavate halduriteni. Rakenduses on selgelt eristuvad kihid, mis jaotuvad järgnevalt: API kiht, funktsionaalne kiht ja andmebaasi kiht. Joonisel 1 on kujutatud diagramm nimetatud kihtidest ning on näidatud kuidas kihid komponentidega suhtlevad.



Joonis 1. Tagasüsteemi kihid ja nende suhtlemise loogika.

API kiht sisaldab endas seda, mis on seotud eessüsteemi rakendusest tulevate päringutega. Kiht sisaldab päringute vastuvõtmispunkte ja mudeleid, mida kiht päringuna sisse laseb ning väljundina väljastab. Kiht sisaldab lisaks veel kaardistajaid (ingl *mappers*), mis kaardistavad sissetulevad ja väljaminevad mudelid API-le mõistetavale kujule, funktsionaalsest kihist tulenevate mudelite alusel.



Funktsionaalne kiht sisaldab endas seda, mis on seotud äriloogikaga ning rakenduse funktsionaalsusega. Kiht võimaldab teha andmete muudatusi, kustutamist ning lisamist. Funktsionaalset kihti kasutab oma tegevusteks API kiht.

Andmebaasi kiht sisaldab endas suhtlusliidest andmebaasiga. Andmebaasi kiht sisaldab endas andmemudeleid, mis on loodud andmebaasi tabelite alusel. Funktsionaalne kiht pärib andmebaasi kihist andmeid oma protsesside jaoks.

Tagasüsteemi ülesanne on eelkõige olla eessüsteemile suhtlusliideseks andmebaasiga.

### **5.1.2 Eessüsteem**

Eessüsteemiks on Angulari raamistikul põhinev rakendus. Rakendus omab suhtlusliidest tagasüsteemiga. Rakenduse osaks on veel vaated ja komponendid, mis moodustavad visuaalse terviku lõpptarbijale. Rakendus on kakskeelne: eesti ja inglise keelne. Rakendust avades kuvatakse vaikimisi eestikeelne sisu. Eessüsteemi loogika ja funktsionaalsus on autori loodud. Disainer lõi kasutajaliidese stiilid ning ka HTML osa, vähesel määral muutis disainer ka funktsionaalsust ja TypeScript osa [4].

### **5.2 Installeerimis- ja käivitamisprotsess**

Rakenduse eessüsteem ja tagasüsteem kompileeritakse ja pakendatakse Maven-iga pakendusfailiks. Pakendusfailist ehitatakse Launch4j käivitusfail. Ehitatud käivitusfail pakendatakse koos Java keskkonnaga installeerimisfailiks, mis võimaldab Windows operatsioonisüsteemil rakendust installida.

Installeerimise tulemusel tekib kasutajale arvutisse rakendus, mida on võimalik käivitada. Rakenduse käivitamisel läheb arvutis käima Java rakendus, mis sisaldab nii taga- kui ka eessüsteemi. Rakenduse eessüsteem käivitub rakenduse kasutaja veebibrauseris. Rakenduse kasutajale avaneb ka aken, millel on üks nupp rakenduse sulgemise funktsionaalsusega. Aken peab olema rakenduse kasutamise ajal avatud. Kui aken suletakse kasutaja poolt, peatab ja sulgeb see rakenduse, kuid ei sulge veebibrauseri vahelehte.

Käesoleva töö raames loodud rakenduse versioon on alla laetav veebist<sup>3</sup>. Hiljem majutatakse rakendus Tartu Ülikooli teadusarvutuste keskses.

---

<sup>3</sup> <https://shorturl.at/insT0>

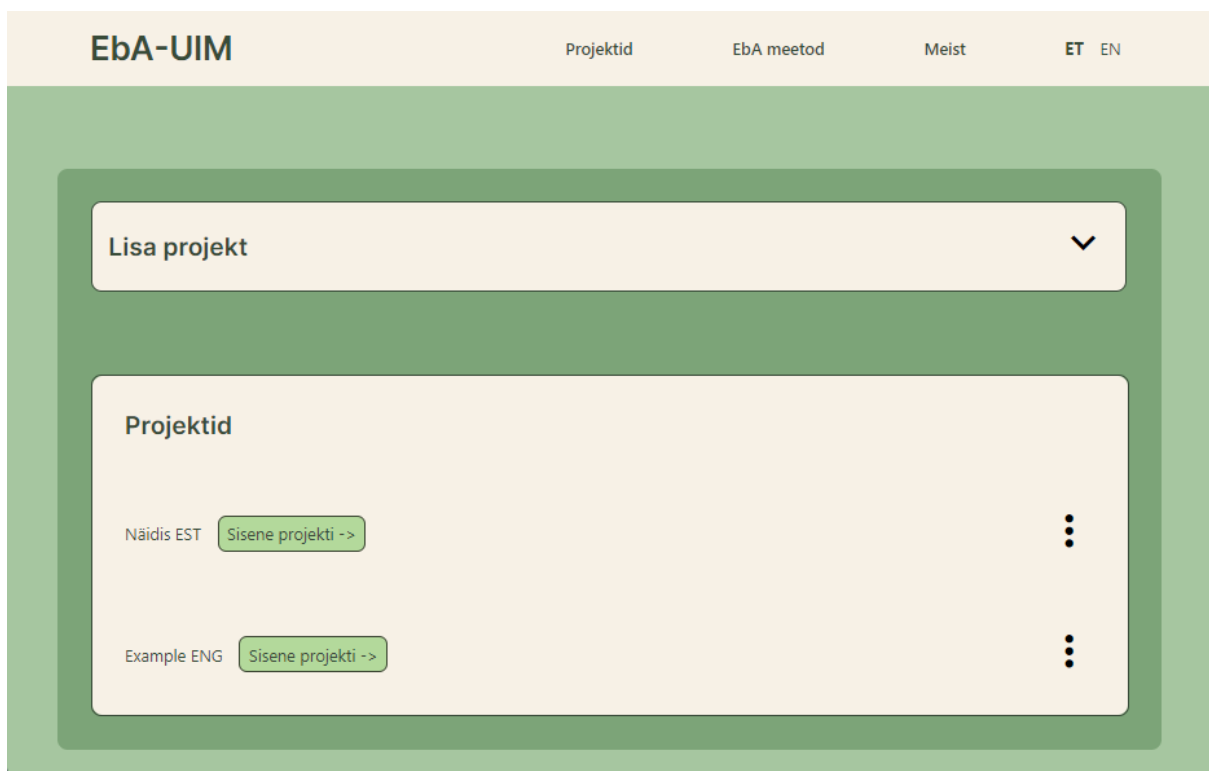
Rakenduse installeerimise käigus võib operatsioonisüsteem teavitada ebaturvalisest rakendusest. See ei tähenda, et rakendus oleks ebaturvaline, vaid rakendusel puudub hetkel ametlik sertifikaat, mis rakenduse Windows operatsioonisüsteemi jaoks turvaliseks tunnistab.

### 5.3 Disaini lahendus

Rakenduse kasutaja näeb rakenduses järgmisi vaateid: projektide halduse vaade, projekti töölehe haldamise vaade, rakenduse kasutusjuhend, tutvustuse vaade. Kõikidele vaadetele tegi kujunduse disainer [4]. Sisutekstit kirjutas klient. Navigatsioon ja kasutajateekonnad otsustati tiimiga ühiselt.

#### 5.3.1 Projekti haldus

Projektide halduse vaates on võimalik projekte nimekirjas vaadata, lisada, kustutada ja nende nime muuta. Vaates on võimalik veel ka navigeerida soovitud projekti sisu vaatesse. Vaate disain ja selle teostus on disaineri tehtud. Joonisel 2 on ekraanitõmmis projekti halduse lehest.



Joonis 2. Rakenduse avavaade: projektide haldus [4]. Ekraanitõmmis rakendusest.

### 5.3.2 Projekti töölehe haldamine

Projekti töölehe halduse vaates on võimalik teha toote või e-teenuse analüüsi EbA meetodit rakendades. Vaates saab lisada ja hallata funktsionaalsuste gruppe, mida esitavad tabeli ülaserivas olevad vahelehed. Vaates on veel ka võimalik lisada ja hallata sidusrühmi, mida on tabeli täitmisel võimalik kasutada.

Tabeli veerupäistel on infonupud, kuhu hiirega peale liikudes avaneb hüpikaken, kus kuvatakse lühike kirjeldus, kuidas antud tulpa kasutada. Tabelis on read ka kahekordselt pesastatud (ingl *nesting*): funktsionaalsuse kirjelduse all saab olla mitu funktsionaalsuse eeltingimuse rida, mille all saab omakorda olla mitu võrreldava situatsiooni rida. Pesastatud ridu saab luua funktsionaalsuse eeltingimuse või -kirjelduse lahtris asuvast menüüst.

Funktsionaalsuse kirjelduse ja eeltingimuse tulbad pikenevad vertikaalselt vastavalt sellele, kui palju alamridu antud tulbal on. Kui ühel eeltingimusel on kolm alamrida, siis eeltingimuse lahter ise katab vertikaalselt kõiki kolme rida. Tabeli real on neli valikuvariantidega lahtrit, mille igas lahtris saab valida nelja variandi vahel. Vastavalt tehtud neljale valikule kuvatakse kasutajatele järgnevates tulpades välja vastus 256 võimalikust variandist ning ka eraldi variant juhuks, kui võrreldava situatsiooni näidet ei ole.

Andmeid muutes või lisades toimub andmete salvestamine jooksvalt andmebaasi. Vaates on võimalik lisada uut rida, vajutades tabeli all olevat nuppu “Lisa uus rida”. Vaate disain ja selle teostus on tehtud disaineri poolt. Joonisel 3 on välja toodud projekti töölehe vaade.

EbA-UIIM

Näidis EST

Projektid

EbA meetod

Meist

ET

EN

Lisa tabeli vahekaart

Lisa uus sidusrühm

Automaatvastaja	Broneering veebis	Meeldetuletus						
Funktsionaalsuse kirjeldus	Sidusrühm	Funktsionaalsuse eeltingimus	Võrreldav situatsioon	Sama sidusrühm?	Sama kontekst?	Eesmärgipärane kasutus?	Lahendusega rahul?	Kirjelus, mil määral on
<div>1</div> <div>Kui juuksurisalongis töö ajal kolme sekundi jooksul keegi telefonile ei vasta, läheb peale automaatvastaja, mille sõnum teatab, et kõik juuksurid on hetkel hõivatud ja soovib võimalusel avada juuksurisalongi kodulehekülge lahe.ee ja teha sealt kaudu broneering või broneeringu muudatus.</div>	Kas	vanemaalisel	mõistavad seda keelt, milles automaatvastaja tekst on sisestatud.	Salongis telefonile vastamine selles keeles, mida kliendid räägivad.	Jah	Osaliselt	Osaliselt	Veidi teises kontekstis rahulolematust selle eeltingimusega, kuid see veidi teistsuguses kontekstis mõningal määral
<div>2</div> <div>Kui juuksurisalongis töö ajal kolme sekundi jooksul keegi telefonile ei vasta, läheb peale automaatvastaja, mille sõnum teatab, et kõik juuksurid on hetkel hõivatud ja soovib võimalusel avada juuksurisalongi kodulehekülge lahe.ee ja teha sealt kaudu broneering või broneeringu muudatus.</div>	Kas	vanemaalisel	märkavad, et automaatvastaja sõnumis on teade, et hetkel ei saa ükski juksur telefoni vastu võtta ja soovitus teha broneering kodulehel lahe.ee.	Automaatvastaja teksti sisu mõistmine nii avaliku- kui erasektori puhul, mis kasutavad automaatvastajat ja suunavad kasutama kodulehte.	Jah	Jah	Jah	Eisneb rahulolematust eeltingimusega, kuid j
<div>3</div> <div>Kui juuksurisalongis töö ajal kolme sekundi jooksul keegi telefonile ei vasta, läheb peale automaatvastaja, mille sõnum teatab, et kõik juuksurid on hetkel hõivatud ja soovib võimalusel avada juuksurisalongi kodulehekülge lahe.ee ja teha sealt kaudu broneering või broneeringu muudatus.</div>	Kas	vanemaalisel	loevad automaatvastaja sõnumist kodulehe aadressi välja korrektselt.	Kodulehe aadressi korrektne mõistmine neil juhtudel, kui juuksurid on aadressi lahe.ee öelnud suuliselt klientidele ja tarnijatele.	Osaliselt	Jah	Osaliselt	Eisneb rahulolematust eeltingimusega, kuid m määral osad
<div>4</div> <div>Kui juuksurisalongis töö ajal kolme sekundi jooksul keegi telefonile ei vasta, läheb peale automaatvastaja, mille sõnum teatab, et kõik juuksurid on hetkel hõivatud ja soovib võimalusel avada juuksurisalongi kodulehekülge lahe.ee ja teha sealt kaudu broneering või broneeringu muudatus.</div>	Kas	vanemaalisel	on valmis häälestamise järel otsima ja avama veebilehte ning tegema seal broneeringut.	Järgmise broneeringu tegemine Salongi kodulehel, kui juuksurid on teeninduse ajal seda	Osaliselt	Osaliselt	Ei	1 Veidi teises kontekstis selle eeltingimusega ra tegutsemise tulemus se veidi teistsuguses kont

Joonis 3. Projekti töölehe vaade [4]. Ekraanitõmmis rakendusest.

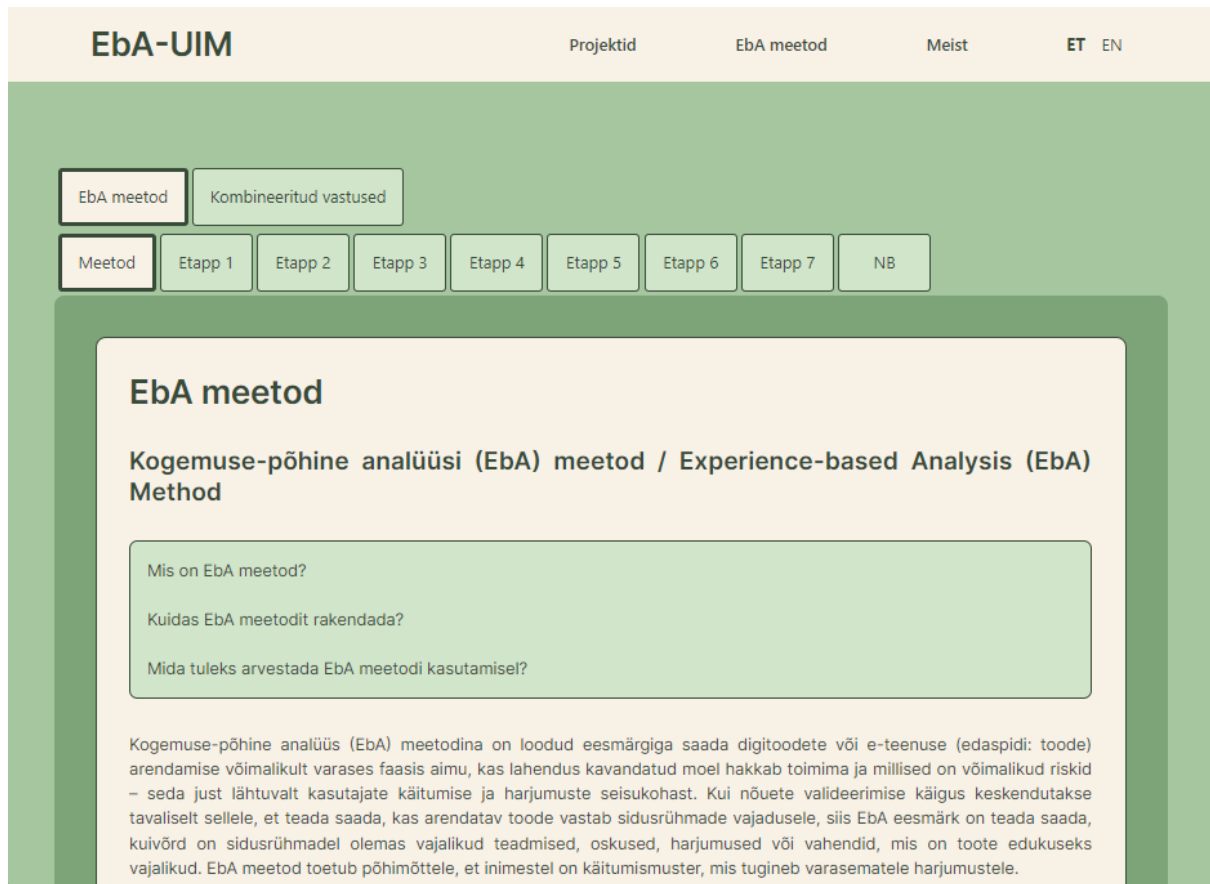
Ridade kustutamiseks on kolm erinevat võimalust. Ridu on võimalik kustutada võrreldava situatsiooni kaupa, mis kustutab vaid selle rea, mis on seotud konkreetset selle situatsiooni väljaga. Veel on võimalik kustutada ridu funktsionaalsuse eeltingimuse kaupa, mis kustutab kogu funktsionaalsuse eeltingimuse alla kuuluvat rea info ja sinna juurde kuuluvad võrreldavad situatsioonid. Kolmandaks on võimalik veel kustutada ridu funktsionaalsuse kaupa, mis kustutab omakorda kõik selle alla kuuluvad funktsionaalsuse eeltingimuse read ja ka selle alla kuuluvad võrreldavate situatsioonide read. Ridade kustutamise kohad on märgitud joonisel 4 punase tooniga.

Automaatvastaja		Broneering veebis		Meeldetuletus					
Funktsionaalsuse kirjeldus	Sidusrühm	Funktsionaalsuse eeltingimus	Võrreldav situatsioon	Sama sidusrühm?	Sama kontekst?	Eesmärgipärane kasutus?	Lahendusega rahul?	Kirjelus, mil määral on	Tee parandused ja tee tegevuskava
12 Järgmine tekst (jäl) kuvatakse kolme sekundi jooksul, kui juuksurid on hetkel hõivatud ja soovibab võimalusel avada juuksurisalongi kodulehekülge lahe.ee ja teha sealt kaudu broneering või broneeringu muudatus.	Kas	vanemaalisel X	Vanemaalisel veebilehel (vanemaalisel) on vahetus	Salongis telefonile vastamine selles keeles, mida kliendid räägivad.	Jah	Osaliselt	Osaliselt	Veidi teises kontekstis rahulolematust selle eeltingimusega, kuid see veidi teistsuguses kont	
13 Kui juuksurisalongis töö ajal kolme sekundi jooksul keegi telefonile ei vasta, läheb peale automaatvastaja, mille sõnum teatab, et kõik juuksurid on hetkel hõivatud ja soovibab võimalusel avada juuksurisalongi kodulehekülge lahe.ee ja teha sealt kaudu broneering või broneeringu muudatus.	Kas	vanemaalisel X	Teatatakse hetkeks on funktsionaalsuse eeltingimus	Automaatvastaja teksti sisu mõistmine nii avaliku- kui erasektori puhul, mis kasutavad automaatvastajat ja suunavad kasutama kodulehte.	Jah	Jah	Jah	Osaliselt	
14 Kui juuksurisalongis töö ajal kolme sekundi jooksul keegi telefonile ei vasta, läheb peale automaatvastaja, mille sõnum teatab, et kõik juuksurid on hetkel hõivatud ja soovibab võimalusel avada juuksurisalongi kodulehekülge lahe.ee ja teha sealt kaudu broneering või broneeringu muudatus.	Kas	vanemaalisel X	loevad automaatvastaja sõnumist kodulehe aadressi välja korrektselt.	Kodulehe aadressi korrektsed mõistmine neil juhtudel, kui juuksurid on aadressi lahe.ee õelnud suuliselt klientidele ja tärnitatele.	Osaliselt	Jah	Osaliselt	Osaliselt	
15 Kui juuksurisalongis töö ajal kolme sekundi jooksul keegi telefonile ei vasta, läheb peale automaatvastaja, mille sõnum teatab, et kõik juuksurid on hetkel hõivatud ja soovibab võimalusel avada juuksurisalongi kodulehekülge lahe.ee ja teha sealt kaudu broneering või broneeringu muudatus.	Kas	vanemaalisel X	on valmis häälestamise järel otsima ja avama veebilehte ning tegema seal broneeringut.	Järgmise broneeringu tegemine Salongi kodulehel, kui juuksurid on teeninduse ajal seda soovitanud.	Osaliselt	Osaliselt	Ei	Ei	

Joonis 4. Ridade kustutamise võimalused [4]. Ekraanitõmmis rakendusest.

### 5.3.3 Rakenduse kasutusjuhend

Valides menüüst aktiivse välja „EbA meetod“ näeb rakenduse kasutaja EbA meetodi detailset kasutusjuhendit. Vaates on vahelehesüsteem, kus iga vaheleht sisaldab endas loogiliselt jaotatud infot. Vaate navigatsioon ja disainilahendus loodud disaineri poolt ning on välja toodud joonisel 5.



Joonis 5. Rakenduse kasutusjuhendi vaade [4]. Ekraanitõmmis rakendusest.

### 5.3.4 Tutvustuse vaade

Valides menüüst aktiivse välja „Meist“ näeb rakenduse kasutaja töövahendi ja EbA meetodi üldist kirjeldust koos rakenduse loojate panuse kirjelduse, kasutusõiguste ja kontaktandmetega. Vaate disaini lahendus on loodud disaineri poolt ning välja toodud joonisel 6.

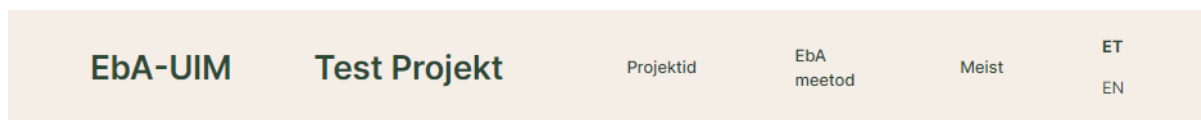


Joonis 6. Rakenduse tutvustuse vaade [4]. Ekraanitõmmis rakendusest.

### 5.3.5 Päis

Rakenduse päis on kasutusel kõikides vaadetes. Päis asub vaate ülemises osas. Päise vasemas servas on suurema fondiga rakenduse nimi ja selle kõrval hetkel valitud projekti nimi. Menüü EbA-UIM viib rakenduse avavaatesse. Projekti nimi annab teada, millise projekti vaates hetkel vaataja asub.

Päises on tekstid linkidena aktiivsed. Linkidena aktiivsete tekstide kaudu on võimalik navigeerida projektide vaatesse, meetodi kirjelduse vaatesse ning tutvustuse vaatesse. Päises on võimalik veel vahetada ka rakenduse kuva keelt vastavalt kas eesti või inglise keelseks, klikkides vastava keele lühendi peal. Päise disain on loodud disaineri poolt ning välja toodud joonisel 7 [4].



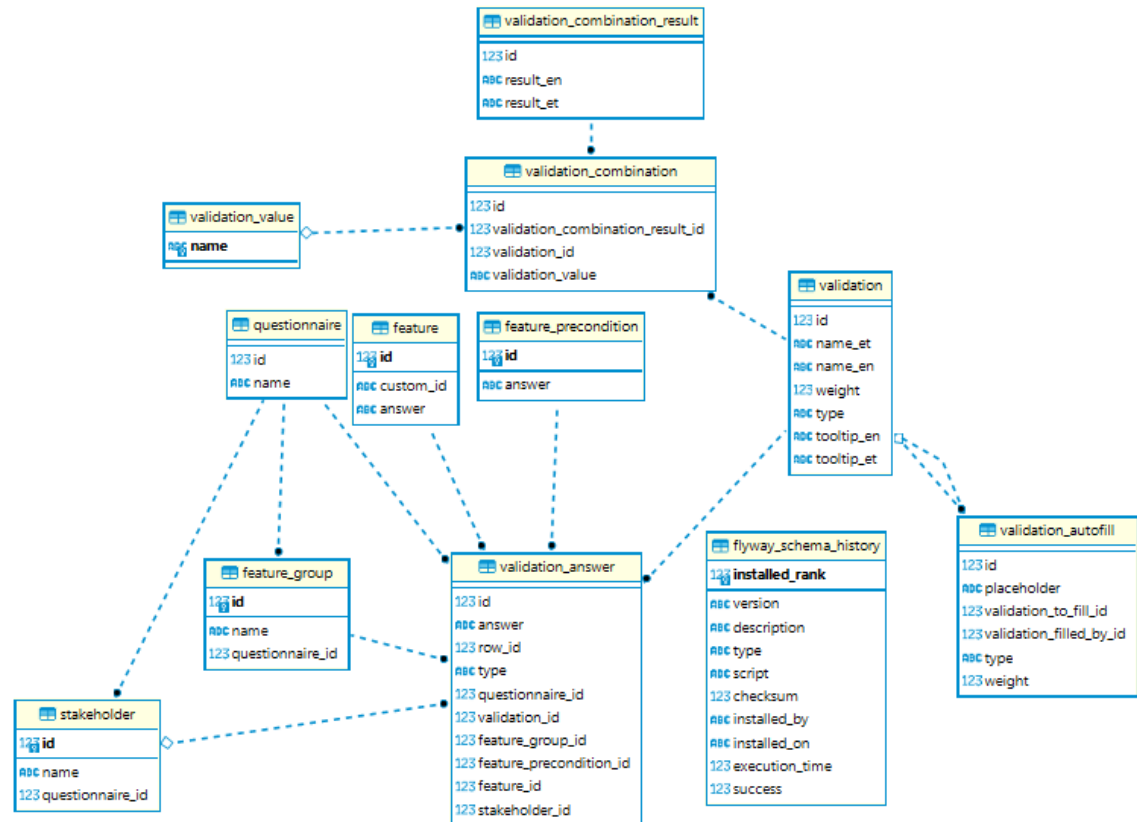
Joonis 7. Rakenduse päis [4]. Ekraanitõmmis rakendusest.

## 5.4 Andmemudel

Andmemudel sisaldab endas kahte tüüpi andmeid. Esiteks sisaldab andmebaas staatilisi andmeid ehk kliendi poolt ette antud väärtuseid, tekste ja väärtuseid, mida on vaja andmete ja eesüsteemi kuvaelementide vahel seose loomiseks. Teiseks sisaldab andmebaas läbi kasutaja toimingute tekitatud andmeid.

Andmemudeli ja andmete läbi skriptide muutmiseks on kasutusel automaatne migreerimisüsteem Flyway. Andmebaasi muutmise skriptid on jaotatud skeemi ja andmete muutmise jaoks. Andmebaasi muutmise skripte käivitab rakenduse nende loomise järjekorras.

Andmemudeli visuaalne skeem on näidatud joonisel 8. Joonisel tähistab iga kast andmebaasi tabelit. Jooned tabelite vahel tähistavad seoseid, kus joone täpiga otsa poolne tabel on viitaja ning joone teises otsas on tabel, millele viidatakse. Eesliidesega *validation* tabelite sisu on automaatselt migreerimissüsteemi poolt andmebaasi sisestatud v.a *validation\_answer*, mille sisend on tulnud kliendi poolt. Eesliidesega *validation* tabeleid ja nendes sisalduvat infot kasutatakse eelkõige projekti töölehe vaates, kuid ka kasutusjuhendi vaates kus on kasutatud illustratsioone. Tabelis *flyway\_schema\_history* hoitakse infot andmebaasi muutmise skriptide kohta. Ülejäänud tabelid täidetakse lõpptarbija sisendi poolt ning kasutatakse EbA meetodi rakendamisel.



Joonis 8. Andmemudel



## 6. Valideerimine

Peatükis antakse ülevaade rakenduse testimise meetoditest ja esitatakse testimise tulemuste kokkuvõte. Testimine jagunes neljaks osaks, milleks olid eeltestimine, kasutatavuse testimise esimene etapp, kasutatavuse testimise teine etapp ja vastuvõtutestimine. Testimise ettevalmistustööd, testimise ja tulemuste analüüsi tegid autor ja disainer koos, et katta ära nii autori kui ka disaineri kummagi skoobis olevate tööde testimisvajadus ühe testimisega ja et testijad saaksid tagasiside anda ühe testimisega. Kasutatavuse ja visuaalsete nõuete osa on leitav disaineri bakalaureusetööst [4].

Kasutatavuse testi valimiks olid inimesed, kes on EbA meetodi ja selle kasutamiseks loodud töövahendi potentsiaalsed kasutajad. Rakendust testiti manuaalselt. Kõik testis osalejad täitsid testi täismahus. Kõikide testide kokkuvõtteks võib öelda, et rakenduse põhifunktsionaalsused toimusid ning testijad olid disainivalikutega rahul. Testülesanded ja tulemused kajastavad vaid antud lõputöö skoopi, st kasutajaliidesega seotud ülesandeid ja tulemusi kajastatakse disaineri bakalaureusetöös [4], siinses töös mainitakse neid vaid ülevaاتlikult.

Kasutatavuse testimise testülesannete koostamisel oli väljakutseks, kuidas tagada, et testijad annaksid tagasiside autori ja disaineri skoopi kuulunud rakenduse arendustöödele, mitte ei hakkaks hindama EbA meetodit ennast. Ühise arutelu tulemusena leiti, et testijatele tuleb ette anda juba loodud näide, mitte paluda testijatel EbA analüüsi läbi viia. Eeltestimise ja kasutatavuse testimise küsimustikud tegid autor ja disainer koos. Disainer tegi sisulised küsimused ning autor rakenduse installeerimise ja maha installeerimisega seotud küsimused.

Esimesena tehti eeltestimine, kus testijaks oli rakenduse klient. Tema valiti eeltestimise etapi testijaks, sest ta oskas kaasa mõelda ja tagasisidet anda nii rakenduse funktsioonide kui ka testülesannete osas. Tegu oli kasutatavuse testi eeltestimisega, et testida rakenduse funktsionaalsust ja küsimustikku. Testülesannete lahendamine ja küsimustele vastamine toimus Google Forms keskkonnas. Testimisele lisandus vabas vormis intervjuu saamaks teada testija üldist arvamust küsimustiku ja rakenduse kasutatavuse kohta. Eeltestimisest selgus, et rakenduse põhifunktsionaalsused toimivad. Eeltestimise tulemusena muudeti esimeses etapis kasutatavat küsimustikku ning parandati rakenduses märgatud mõningaid visuaalseid probleeme.

Järgmiseks viidi läbi kasutatavuse testimine potentsiaalsete lõppkasutajatega. Testimise ülesanded ja küsimustik edastati testijatele Google Forms keskkonnas. Küsimustik sisaldas rakenduse installeerimise juhiseid ja rakenduse allalaadimise linki. Testide sisuks oli rakenduse

põhifunktsionaalsuste katsetamine ning disaini hindamine. Testimisele lisandusid vabas vormis intervjuud saamaks teada testijate üldiseid arvamusi rakenduse paigaldamise ja kasutatavuse kohta.

Kasutatavuse testimise esimesest etapist võttis osa kaks inimest. Testimisest selgus, et rakenduse põhifunktsionaalsus toimib, kuid esines arusaamatusi rakenduse paigaldamisel ning rakenduse mõistmisel. Kasutatavuse testimise esimese etapi järel parendati testimiseks kasutatavat küsimustikku, mille muudatused sisaldasid endas küsimustiku üldistamist, et suunata testijaid lähenema ülesannetele rohkem abstraktsemalt, selmet juhust jälgida. Testimise tulemusena parandati veel ka rakenduses kuvatavaid tekste. Eeltestimise ja esimese etapi küsimustik on leitavad Lisast I.

Kasutatavuse testimise teises etapis kasutati samu teste ja meetodeid, mis esimeses etapis, kuid täiendatud kujul. Põhiline muudatus esimese ja teise etapi vahel on see, et testijad pidid rakendust kasutama iseseisvamalt kui testimise esimeses etapis. Testimise teisest etapist võttis osa kolm inimest. Teise etapi tulemus kinnitas, et rakenduse põhifunktsionaalsused töötavad. Veel selgus, et testijatele rakenduse disain üldiselt meeldib ning et testi täites ei tekkinud suuremaid arusaamatusi. Teise etapi küsimustik ja testimiseks kasutatav andmestik on leitavad Lisast II ja III.

Vastuvõtutestimisel testiti kõik kasutuslood, mis on loetletud peatükis 2.2. Testimiseks vormistati kasutuslood „antud-kui-siis“ (ingl *given-when-then*) malliga [20]. Selle malli järgi kirjeldatakse testijale sammus „Antud“, mis peab olema talle testi sooritamiseks ette antud. „Kui“ sammus kirjeldatakse testijale, mida ta peab tegema. „Siis“ sammus kirjeldatakse, mis peab juhtuma peale testija „Kui“ sammu sisendit. Enne lõpptestijale suunamist tegi kõikidele „antud-kui-siis“ malliga loodud kasutuslugudele kontrolli rakenduse arendustiimi disainer. Vastuvõtutesti lõpptestijaks oli klient. Klient andis igale „antud-kui-siis“ etapile tagasiside, kas a) läbitud, b) ei õnnestunud läbida, c) ebakindel, d) blokeeritud, e) määratlemata. Lisaks etappidele andis testija sama valiku põhjal tagasiside kogu kasutusloole. Vastuvõtutestimisel leiti kriitilised vead, mis pärssisid rakenduse tööd: jõudlus, ridade kustutamine, kuva keele vahetamine. Vastuvõtutestimise tulemusena leitud vead parandati. Vastuvõtutestimise küsimustiku malli vormistas disainer [4]. Autor koostas installeerimise, maha installeerimise ja rakenduses staatiliste tekstide muutmise küsimused ning disainer rakenduse kasutamise küsimused.

## 7. Väljakutsed

Käesolevas peatükis käsitletakse erinevaid väljakutseid, mis arendusprotsessi käigus tekkisid.

Üks väljakutse oli seotud rakenduse visualiseerimisega. Autoril puudus eelnev kogemus visuaalse disaini ning selle teostamisega. Projekti algusfaasis oli plaanis ehitada funktsionaalne rakendus ilma disainielementideta. Disaini osa oli plaanis jätta hilisemaks ajaks peale funktsionaalse osa valmimist. Projekti keskpaigas liitus tiimiga disainer. Disaini arenduse käigus tekkisid ka funktsionaalsed muudatused. Tekkinud muudatused ei olnud osaliselt kooskõlas ehitatud baasfunktsionaalsusega, mis tähendas seda, et tuli baasfunktsionaalsust muuta ning panna toimima koos disainiga. Kui disainer oleks olnud kaasatud kohe projekti algusfaasis, oleks funktsionaalsus kohe alguses lahendatud teisiti ning vastavuses disainiga. Tagantjärele võib öelda, et teave disainilahenduse kohta on oluline projekti algusfaasis, kuna rakenduse ülesehitus sõltus ka disainist.

Väljakutseks oli ka disainist tuleneva ja funktsionaalse koodi kokku viimine ja ühtlustamine lähtekoodi tasandil. Tekkis olukordi, kus funktsionaalsus oli arendusega edasi liikunud disaini protsessi arvestamata, kuid oli ka vastupidiseid olukordi. Disainer ja autor kooskõlastasid omavahelise kommunikatsiooni protsessi, mille tulemusena suheldi tihedamalt seoses valmivate arendustega ning üldine arendus liikus edasi väiksemate sammudega, mis lahendas tekkinud probleemi.

Kõige rohkem väljakutseid tekkis eelkõige eessüsteemi arendades ning installeeritavat rakendust ehitades. Üks kriitilisemaid hetki oli, kui peale uue versiooni installeerimist ei uuenenud lõpptarbija veebilehitsejas kasutajaliides. Probleemi põhjustas kasutajaliidese hoiustamine veebilehitseja vahemälu. Lahenduseks oli seadistada rakenduse ehitamisel kasutajaliides nii, et veebilehitseja arvaks, et tegu on täiesti uue kasutajaliidesega.

Rakenduse mittefunktsionaalseid ja funktsionaalseid nõudeid arvesse võttes tekitasid osad – üldpildis rakenduse töötamise ja kasutajasõbralikkuse suhtes mitteolulised nõuded – suurt ajakulu. Näiteks kulus kasutajaliideses tabeli funktsionaalsuse arendamisele, et horisontaalselt kerides oleksid tabeli esimesed tulbad staatiliselt kerides paigal, vähemalt päeva jagu arendust. Sarnased probleemid ning ajakulu esines ka teiste nõuete täitmisel, mis autori arvates oleks tehnoloogiat paremini oskaval inimesel kordades kiiremini valminud.

Projekti üks keerukamaid etappe oli eessüsteemi ja tagasüsteemi rakenduste ühendamine installeeritavaks rakenduseks, mis töötaks Windows operatsioonisüsteemiga arvutites ilma

täiendavate lisasammudeta. Autoril puudus sellega varasem kogemus. Installeeritava rakenduse saamiseks kasutas autor Launch4j ja Inno Setup tehnoloogiaid. Neid tehnoloogiaid ei ole autor varem kasutanud. Töötav lahendus tekkis tehnoloogiate dokumentatsiooni järgimise ja katsetamise meetodil.

Väljakutseks oli ka ettenähtud ajaraamis püsimine, kuna arendusvajaduste muudatused ületasid planeeritud ajamahtu. Ajamahu ületamise üheks põhjuseks oli kindlasti see, et projekti alguses jäi detailselt määratlemata bakalaureusetöösse mahtuvad nõuded ja jäi arvutamata neile kuluv aeg. Arenduste järjekord ja prioriteedid kinnitati hiljem, kui töödega oli juba alustatud. Arendustiimis teadvustati ajaraamist kinnipidamise olulisust ning kui jõuti testimise etappi, siis testimiselt saadud tagasiside põhjal koostatud arendusvajadused prioritseeriti tiimi ühise arutelu käigus ja muudatused viidi sisse vaid kõige prioriteetsematele vajadustele. Kokkuvõttena saab kinnitada, et projekte planeerides peab realistlike ootuste loomiseks nõuded valima etappi selliselt, et tööd mahuksid ettenähtud ajaraami.

## 8. Jätakuarendused

Käesoleva töö skoopi sai valitud ja teostatud osa kogu rakenduse nõuetest. Jätakuarendustena on kavandatud järgmised tegevused:

1. Projekti eksportimine Exceli tabelina.
2. Tabeli tulpade taustavärvi muutmise võimekus.
3. Prügikastisüsteemi sisseviimine kustutamisele, mis sisaldab endas kasutaja poolt kustutatud sisu taastamist.
4. Tabeli tulpade peitmine kasutaja sisendi läbi.
5. Tabelis olulisemate ridade esiletoomine.
6. Rakenduse töökindluse parendamine.
7. Koodi refaktoreerimine.
8. Veebilehitseja lehe värskendusfunktsionaalsuse sisseviimine.
9. Tabeli filtreerimine ja sorteerimine.
10. Taga- ja eessüsteemi automaattestimine ühiktestidega.
11. Kohandamine iOS operatsioonisüsteemidele.

## 9. Kokkuvõte ja järeldused

Bakalaureusetöö eesmärgiks oli luua kliendi sisendi ja prototüübi alusel rakendusprogramm EbA meetodile. Töö raames loodud rakendusprogramm hõlbustab EbA meetodi kasutamist. Paralleelselt antud lõputööga valmis ka rakenduse disain, mille teostas disaineri rolli täitnud kaastudeng.

Töö käigus õppis autor, kuidas luua installeeritavat rakendust ja kuidas koostada kasutajaliideses keerulisi tabelivaateid. Olulisteks õppemomentideks olid ka disaineri ja arendaja koostöö olulisus ning kliendi nõuete mõistmise ja rahuldamise tähtsus.

Valminud rakenduse eessüsteem on loodud Angular raamistiku abil. Tagasüsteem loodi Spring raamistiku abil ning rakendus kasutab SQLite andmebaasi. Rakendust pakutakse installeeritava rakendusena.

Rakendusele teostati ka erinevaid manuaaltestimisi. Testimises osales kokku kuus testijat. Testimise tulemusena selgus, et rakenduse põhifunktsionaalsused toimivad ning rakendus on üldiselt inimeste jaoks arusaadav ja mugav kasutada.

Rakenduse edasise kasutamise ja testimise käigus avastati jooksvalt veel tehnilisi vigu, millest kriitilisemad said ka parandatud. Rakenduse töökindluse tagamiseks tulevikus on vaja kirjutada ühiktestid. Peatükis 8 on loetletud jätkuarendused, mida on vaja täiendavalt arendada, et parandada rakenduse töökindlust ja lisada rakendusele veel puuduvad funktsionaalsused. Rakendust on võimalik alla laadida Google Drive veebiportaalist<sup>4</sup>.

---

<sup>4</sup> <https://shorturl.at/insT0>

## Viidatud kirjandus

- [1] Piirisild A. Tool for the Experience-based Analysis (EbA) method, 2024.  
<https://datadoi.ee/handle/33/594> (22.04.2024)
- [2] Piirisild A., Perandrés Gómez A., Taveter K. A New Usability Inspection Method: Experience-Based Analysis. *Requirements Engineering: Foundation for Software Quality*, 2024. [https://doi.org/10.1007/978-3-031-57327-9\\_5](https://doi.org/10.1007/978-3-031-57327-9_5) (15.05.2024)
- [3] Ralyté J., Bork D., Jeusfeld M. A., Kirikova M., Stirna J. Panel Discussion: How to Build a Perfect Enterprise Modeling Method. *In Proceedings of the Forum at Practice of Enterprise Modeling*, 2021. <http://hdl.handle.net/20.500.12708/55643> (15.05.2024)
- [4] Kreinin, I. Kasutajaliidese disaini loomine „Kasutatavuse kogemuspõhise analüüsi meetodi“ rakendusele. Tartu Ülikooli arvutiteaduse instituudi bakalaureusetöö. 2024.
- [5] Cybernetica AS. Andmekaitse ja infoturbe portaal. <https://akit.cyber.ee/> (22.04.2024)
- [6] SQLite. <https://www.sqlite.org/> (03.07.2023)
- [7] Oracle. Java Developer's Guide. <https://docs.oracle.com/en/database/oracle/oracle-database/19/jjdev/Java-overview.html> (07.02.2024)
- [8] VMware Tanzu. Spring Framework. <https://spring.io/projects/spring-framework> (03.07.2023)
- [9] Apache. Maven. <https://maven.apache.org/> (03.20.2024)
- [10] Microsoft. TypeScript. <https://www.typescriptlang.org/> (22.04.2024)
- [11] Google. Angular. <https://angular.io/guide/architecture> (22.04.2024)
- [12] Sourceforge. Launch4j. <https://launch4j.sourceforge.net/> (25.03.2024)
- [13] Sourceforge. Launch4j Executable Wrapper. Download Statistics  
<https://sourceforge.net/projects/launch4j/files/stats/timeline> (25.03.2024)
- [14] Laan M., Russell J. Inno Setup. <https://jrsoftware.org/isinfo.php> (20.03.2024)
- [15] Laan M., Russell J. Inno Setup Downloads. <https://jrsoftware.org/isdl.php> (20.03.2024)
- [16] SpringAngularDemo. <https://github.com/marco76/SpringAngularDemo> (20.03.2024)
- [17] VMware. Web MVC framework. <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html> 20.03.2024)

- [18] Chacon S., Long J. Git. <https://git-scm.com/> (01.04.2024)
- [19] Hammant P., Smith S. Trunk Based Development: Introduction.  
<https://trunkbaseddevelopment.com/> (01.04.2024)
- [20] Pandit P., Tahiliani S., Sharma M. Distributed agile: Component-based user acceptance testing. *Symposium on Colossal Data Analysis and Networking*, 2016.  
<https://ieeexplore.ieee.org/document/7570913> (15.05.2024)



## **Lisad**

### **I Küsimustik**

Lisa I sisaldab eeltestimise ja kasutatavuse testimise esimese etapi küsimustikku bakalaureusetöö skoobi osaga. Küsimustiku sissejuhatus ja testimise I osa küsimustik on sama, mis disaineril [4], kuna testimist valmistati ette ja viidi läbi ühiselt. Testid on koostatud inglise keeles, et rakendust saaks testida ka eesti keelt mittevaldavad testijad.

### **EbA-UIM tool usability testing**

We are glad to welcome You as a participant in our usability testing. The purpose is to gather insightful data about EbA-UIM tool application potential users' interactions and experiences with the application.

This project is undertaken by two Bachelor students from the Computer Science Unit at the University of Tartu, Estonia. Our approach involves several stages:

1. Initially, we would like to present You with various usability-related tasks to assess Your interaction with our application.
2. Following that, we seek Your feedback on the overall visual experience of the application.
3. Finally, we have prepared a brief questionnaire related to our application, aiming to gather Your insights and opinions.

Your participation in each stage of this process is highly appreciated and will contribute significantly to the refinement and improvement of our application, Eba-UIM tool.

The application to download itself is an installer, which installs the EbA method. Since this installer is not created with a paid identification key window will prompt the installer as an unsafe application. This is not the case. While doing the test scenario where You have to install the application please click “more info” on the screen that pops up and then You are able to install the application.

The application size is about 432 megabytes and will be in its entirety installed on the path You select during installation. After installation the folder selected during installation will also consider an uninstaller from which You can uninstall the application.

NB! The uninstaller will not uninstall the database file created for the application, the database file will still be in the same folder selected during installation, this file for now must be deleted manually.

After installation a shortcut to launch the application (if ticked during the installation process) will be on the users Desktop. If the user chooses not to create a shortcut for Desktop, the application must be launched from the folder where it was installed.

After launching the application it will open a new tab in Your default web browser which also is the main component of the application. It will also open a separate window in a separate application which contains the functionality to close the application process.

To close the application click the button on the screen which was opened and also close the browser tab. The opened browser tab is the application where most of the testing is done.

Setting description: Windows OS

Known issues: Sometimes when deleting rows, the table might visually break. To fix this please navigate to <http://localhost:8080/> .

TasksThe first part of the testing consist of five different tasks that You need to complete in the EbA-UIM tool. Also, feel free to explore the application.

## **Scenario 1**

Download the installer

Installer can be downloaded from

here:[https://drive.google.com/file/d/15zDD\\_SvBzyZpoZo9XcFTWqkN2Q7XiWly/view?usp=sharing](https://drive.google.com/file/d/15zDD_SvBzyZpoZo9XcFTWqkN2Q7XiWly/view?usp=sharing)

Open the installer and install the application. Also create a desktop shortcut using the installer and write down the path where You installed the application as it isrequired in a later case.

Find the created shortcut on Your Desktop and open the application. Were You able to complete this task?

If You were unable to complete the first task, what do You think made it difficult or prevented its completion?

On a scale of 1 to 5 how would You rate this task?

## **Scenario 2**

Firstly choose Your preferred application language - Estonian or English Try to create two new projects and name them how You like.

Change one project name to "EbA-UIM 2024" and delete the other project.

Were You able to complete this task?

If You were unable to complete the second task, what do You think made it difficult or prevented its completion?

On a scale of 1 to 5 how would You rate this task?

### Scenario 3

Now You have a project named "EbA-UIM 2024" . Find a way how to apply the EbA method to this feature:

Feature: "One day before the service, the system sends a reminder to the e-mail specified by the customer with information about the service."

Your staholders are "elderly customers".

You have a precondition that they "remember and don't forget their appointment if they haven't given an email address."

You remember a situation about them that they "remember appointments made by phone if no email address has been given."

The stakoholders, context are same in this situation and the use was purposeful but the stakeholders were partly satisfied.

If You do everything right You should have "There is somewhat dissatisfaction with this precondition, but yes, elderly customers remember and don't forget their appointment if they haven't given an email address." as an answer. Hint: You need a feature group to apply the method to features.

Were You able to complete this task?

If You were unable to complete the third task, what do You think made it difficult or prevented its completion?

On a scale of 1 to 5 how would You rate this task?

### Scenario 4

Now You have completed Your first situation analysis.

You have a new precondition that the stakeholders "are checking their email."

Add this precondition the same feature. The stakeholders are the same but there is no comparable situation. Figure out how to still get an anwer to this precondition.

Were You able to complete this task?

If You were unable to complete the fourth task, what do You think made it difficult or prevented its completion?

On a scale of 1 to 5 how would You rate this task?

## **Scenario 5**

Analyse another feature "After the customer has chosen a service (women's, men's, or children's hairdresser), the system asks for the hair length (short, shoulder-length, long).".

The stakeholders stay the same and the precondition is "After the customer has chosen a service (women's, men's, or children's hairdresser), the system asks for the hair length (short, shoulder-length, long)."

This time You have two situation examples. Find a way how to add another situation to the same feature and precondition. You do not need to insert any situation texts.

Were You able to complete this task?

If You were unable to complete the fifth task, what do You think made it difficult or prevented its completion

On a scale of 1 to 5 how would You rate this task?

What difficulties did You encounter during the testing?

How could we improve the usability?

## **Instructions for deleting the application**

Open the path where You installed the application and uninstall the application by running "unins000".

Feel free to ask for help.

Were You able to complete this?

## II Teise etapi küsimustik

Lisa II sisaldab kasutatavuse testimise teise etapi küsimustikku bakalaureusetöö skoobi osaga. Testid on koostatud inglise keeles, et rakendust saaks testida ka eesti keelt mittevaldavad testijad. Küsimustiku sissejuhatus ja testimise I osa küsimustik on sama, mis disaineril [4], kuna testimist valmistati ette ja viidi läbi ühiselt.

### EbA-UIM tool usability testing

We are glad to welcome You as a participant in our usability testing. The purpose is to gather insightful data about EbA-UIM tool application potential users' interactions and experiences with the application.

The motivation for creating the Experience-based Analysis (EbA) method is to make the development of digital products and services more cost effective, since EbA method provides an early indication in digital product or e-service development of whether the solution will function efficiently in terms of stakeholders' behavior and habits, and identify potential risks that may arise.

The EBA-UIM tool was created to make using the Experience-based Analysis (EbA) method easier. The tool helps structure the information and automatically displays the descriptions predicting the usability of the product/e-service under development.

Known issues:

Sometimes when deleting rows, the table might visually break. To fix this please navigate to <http://localhost:8080/>.

### Tasks

The first part of the testing consist of three different tasks that You need to complete in the EbA-UIM tool. Also, feel free to explore the application.

### Scenario 1

Download the installer

Installer can be downloaded from

here:<https://drive.google.com/file/d/1zHvqSbpsKlASFEnTaB7Lxv3pDI8Ppf4C/view?usp=sharing>

Open the installer and install the application. Also create a desktop shortcut using the installer and write down the path where You installed the application as it is required in a later case.

Find the created shortcut on Your Desktop and open the application. Were You able to complete this task?

If You were unable to complete the first task, what do You think made it difficult or prevented its completion?

On a scale of 1 to 5 how would You rate this task?

## **Scenario 2**

Firstly choose Your preferred application language - Estonian or English Try to create two new projects and name them how You like.

Change one project name to "EbA-UIM 2024". Then delete the other project.

Were You able to complete this task?

If You were unable to complete the second task, what do You think made it difficult or prevented its completion?

On a scale of 1 to 5 how would You rate this task?

## **Scenario 3**

Now You have a project named "EbA-UIM 2024". Go to this project.

Then go to this link to access two Eba method example uses and choose if you would like to use English or Estonian version.

[https://drive.google.com/drive/folders/1\\_XoymuBLDlLqdmUFVnk0bG1TZr0R3eg6?usp=sharing](https://drive.google.com/drive/folders/1_XoymuBLDlLqdmUFVnk0bG1TZr0R3eg6?usp=sharing)

Then try to get similar result using the Eba-UIM tool. Were You able to complete this task?

If You were unable to complete the third task, what do You think made it difficult or prevented its completion?

On a scale of 1 to 5 how would You rate this task?

## **Instructions for deleting the application**

Open the path where You installed the application and uninstall the application by running "unins000".

Feel free to ask for help.

Were You able to complete this?

### III Andmed testimiseks

Andmeid kasutasid testijad kasutatavuse testimise teises etapis. Testijad pidid kandma andmed testi käigus rakendusse. Andmed testimiseks koostas klient.

	Funktsionaalsus	Funktsionaalsuse eeltingimused, mille kehtivust hakatakse kontrollima		Võrreldava situatsiooni näide	Prioritseeri	Sama sidusrühm?	Sama kontekst?	Eesmärgipärane kasutus?	Lahendusega rahul?	Prognoos funktsionaalsuse (feature) kasutatavuse kohta lähtuvalt analüüsitud oludest			Prognoosi koond	Tee järelused ja loo tegevuskava	
	Jagage toode/teenus väiksemateks iseseisvateks funktsionaalsusteks ja sisestage siia nende kirjeldused: nt. võited selle kohta, mida kasutajad saavad teha, kasutuslood, kasutusjuhendid, stsenaariumid, eesmärgimudelid funktsionaalsed eesmärgid, protsesside kirjeldused, teekonnakaardistus, ärianalüüsi dokumendi info vm, mis kirjeldab kavandatavat lahendust ja kus on	Kes on funktsiooniga seotud või funktsiooni märkimisväärselt mõjutavad sidusrühmad? Milliseid olulisi teadmisi, hoiakuid, oskusi, teadlikkust, emotsioone, vahendeid, kompromissivõimaldusi ja tegevusi sellel sidusrühmal selle spetsiifilise funktsiooni raames eeldatakse, et kirjeldatud funktsionaalsus tooks tulemust. Kirjeldage siia, mida sidusrühm peab eeltingimuseks tegema, teadma või suutma teha. Mõelge ka sellistele eeldustele, mis ei ole toote/teenusega otseselt seotud, kuid mida ta oluliselt mõjutab. Sõnastage eeldused eesmärgi		Eeltingimused nimetatud tegevused, oskused, suhtumine, vahendid, seadmete kasutamine jms väljenduvad mingi situatsiooni mingisuguses käitumises. Leidke võimalikult sarnane situatsioon, mida sidusrühm on varem kogenud ja kus ilmneb samasugune sidusrühma toiming, nagu on kirjeldatud eeltingimuses.	Kui ühe eeltingimuse kohta olete toonud mitu näidet, palun valige nendest kõige asjakohasem näide.	Kas eelduses nimetatud sidusrühm on sama, keda on peetud silmas näites kirjeldatud situatsioonis?	Kas eelduses tood kontekst on sama, mida on peetud silmas näitena kirjeldatud situatsioonis? Nt kas aeg, tehnoloogia, kasutuseesmärk, vahendid, sotsiaalne,	Kas näitena toodud situatsioonis on sidusrühm tegutsenud ettenähtud ja oodatud viisil?	Kas sidusrühm selles näiteks toodud situatsioonis oli selle lahendusega rahul? Rahulolu tähendab (sagedast) kasutamist ilma kaebusteta.	EBA meetod annab kombineeritud vastuse, mis esitab kasutatavuse olukorda lähtuvalt eelnevalt kirjeldatud eeldusest, sh vaatluse alla võetud tingimustes. Kombineatsioon saadakse järgmiselt: neljale küsimusele antud vastuste põhjal kuvab EBA meetod olukorra kirjelduse (vt "KPA meetod" ptk "Kombineeritud vastus"), millele liidetakse eespool sisestatud sidusrühm koos eeldusega.			Eelnev kolmeosaline lause liidetakse üheks lauseks.	Saadud vastuse põhjal otsustage, kas võrdluseks toodud näitesituatsioonist tuletatud eeldused on lahenduse edukaks toimumiseks piisavad või kui mitte, siis mida teha, seda parandada?	
ID	Funktsionaalsuse kirjeldus	Kas	Sidusrühm	Funktsionaalsuse eeltingimus	Võrreldav situatsioon	Prioritseeri	Sama sidusrühm?	Sama kontekst?	Eesmärgipärane kasutus?	Lahendusega rahul?	Combination of answers (from page "H-K combinations")	Stakeholders (from column D)	Features precondition (from column E)	Prognoosi koond	Tee järelused ja loo tegevuskava.
10	Üks päev enne teenust saadab süsteem kliendi märgitud e-mailile meeldetuletuse koos teenuse infoga.	Kas	vanemaealised	teavad, miks on oluline õige juuste pikkuse määramine.	Varasemalt telefoni teel juuste pikkuse õigesti määramine.		Jah	Osaliselt	Osaliselt	Osaliselt	Veidi teises kontekstis esineb rahulolematust selle eeltingimusega, kuid selles veidi teistsuguses kontekstis mõningal määral	vanemaealised	teavad, miks on oluline õige juuste pikkuse määramine.	Veidi teises kontekstis esineb rahulolematust selle eeltingimusega, kuid selles veidi teistsuguses kontekstis mõningal määral vanemaealised teavad, miks on oluline õige juuste pikkuse määramine.	Usume, et kliendid mõistavad selle olulisust, aga kui pole võimalik aru saada, mida lausega mõeldakse, siis sealt tekibki segadus. Lause tuleb täpsemaks muuta.
		Kas	vanemaealised	teavad, mis pikkus on lühike, õlani või pikk.	Telefoni teel broneerides on pidanud teadma juuste hetke pikkust.	1	Jah	Jah	Osaliselt	Osaliselt	Esineb rahulolematust selle eeltingimusega, kuid mõningal määral	vanemaealised	teavad, mis pikkus on lühike, õlani või pikk.	Esineb rahulolematust selle eeltingimusega, kuid mõningal määral vanemaealised teavad, mis pikkus on lühike, õlani või pikk.	Lisame joonise, kus on näidatud juustepikkuse vahemikud.
					Varasemalt veebilehel broneeringut teinud kliendid on valinud teenusliikide seast juuste õige pikkuse.	2	Osaliselt	Jah	Osaliselt	Osaliselt	Esineb rahulolematust selle eeltingimusega, kuid mõningal määral osad	vanemaealised	teavad, mis pikkus on lühike, õlani või pikk.	Esineb rahulolematust selle eeltingimusega, kuid mõningal määral osad vanemaealised teavad, mis pikkus on lühike, õlani või pikk.	
		Kas	vanemaealised	käivad vaatamas e-maili.	Näidet pole						Kuna ühtegi näidet pole võrdluseks tuua, ei ole teada, kas	vanemaealised	käivad vaatamas e-maili.	Kuna ühtegi näidet pole võrdluseks tuua, ei ole teada, kas vanemaealised käivad vaatamas e-maili.	E-mail on hea meeldetuletaja. Vaatame, juhul kui ainult telefoni numbri andjatel tekib palju unustamisi, siis proovime leida alternatiivse lahenduse.
		Kas	vanemaealised	on meeles ja ei unusta broneeringu aega, kui nad pole andnud e-maili aadressi.	Vanemaealised pole unustanud broneeringut, mis on tehtud telefoni teel ja e-maili aadressi pole antud.		Jah	Jah	Jah	Osaliselt	Esineb rahulolematust selle eeltingimusega, kuid jah,	vanemaealised	on meeles ja ei unusta broneeringu aega, kui nad pole andnud e-maili aadressi.	Esineb rahulolematust selle eeltingimusega, kuid jah, vanemaealised on meeles ja ei unusta broneeringu aega, kui nad pole andnud e-maili aadressi.	

Joonis 9. Andmed kasutatavuse testimise teiseks etapiks

## Litsents

### **Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks**

Mina, Karl Olaf Kuldmaa,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose EbA meetodi rakendusprogrammi arendus, mille juhendajad on Kuldar Taveter ja Anu Piirisild, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

*Karl Olaf Kuldmaa*

**13.05.2024**