

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Rando Leppik

VEEBIRAKENDUSE LOOMINE ETTEVÕTTELE TÖÖLEIDJA

Bakalaureusetöö (9 EAP)

Juhendaja: Vambola Leping

Tartu 2021

Veebirakenduse loomine ettevõttele Tööleidja

Lühikokkuvõte:

Bakalaureusetöö käsitleb veebirakenduse loomist ettevõttele Tööleidja. Töö sisaldab sarnaste Eesti portaalide analüüsi, loetelu rakenduse nõuetest, ülevaadet kasutatud tehnoloogiatest ja töövahenditest ning tarkvara arhitektuurist, rakenduse loomise protsessi kirjeldust ja tulemuse analüüsi.

Võtmesõnad:

Tööleidja, veebirakendus, veebileht, React, Redux, CSS-in-JS, Node.js, Express.js, MongoDB

CERCS:

P175 Informaatika, süsteemiteooria

Creating a web application for Tööleidja

Abstract:

This Bachelor's thesis disserts the creation of a web app for the company Tööleidja. The thesis includes an analysis of similar platforms created in Estonia, descriptions of the requirements of the application, an overview of tools used to create it, and the architecture of the software, the process of creating the application and analysis of the result.

Keywords:

Tööleidja, web application, website, React, Redux, CSS-in-JS, Node.js, Express, MongoDB

CERCS:

P175 Informatics, systems theory

Sisukord

1. Sissejuhatus.....	4
2. Mõisted ja terminid	5
3. Sarnased tarkvaralahendused Eestis	6
4. Arendusnõuded	8
4.1 Funktsionaalsed nõuded	8
4.2 Mittefunktsionaalsed nõuded	9
5. Kasutatud tehnoloogiad	10
5.1 Kliendipoolsed tehnoloogiad.....	10
5.1.1 React	10
5.1.2 Alternatiivsed <i>front end</i> raamistikud	10
5.1.3 CSS-in-JS.....	11
5.2 Serveripoolsed tehnoloogiad	11
5.2.1 Node.js	11
5.2.2 Express.....	12
5.2.3 Socket.IO	12
5.2.4 Andmebaas	12
6. Rakenduse arhitektuur ja arendus	14
6.1 Kasutajaliides	14
6.2 Server ja andmebaas	16
6.3 Töövahendid.....	20
7. Tulemuse ülevaade ja analüüs	23
8. Kokkuvõte.....	34
9. Viidatud kirjandus.....	35
Lisad.....	37
Litsents	43

1. Sissejuhatus

Ootamatult maailma tabanud koroonaviiruse pandeemia tagajärjel jäi 2020. aastal nii globaalses mastaabis [1], kui Eestis [2] hulk ühiskonnast ettenägematutel põhjustel töötuks. Uuest paratamatust reaalsusest tulenevalt kogusid enam kui varem populaarsust alternatiivsed töösuhtevormid, sealhulgas vabakutselisus [3]. Lisaks saab üha rohkem inimesi oma sissetuleku platvormitööst – tööst, mida hallatakse digitaalselt ja tehakse endale sobivatel tingimustel virtuaalses või füüsilises keskkonnas [4]. Platvormitöö leviku kasv, eelkõige noorema põlvkonna seas, viitab iseseisvuse väärtustamisele, soovile olla oma aja peremees.

Seoses eelnimetatud muutuste ja trendidega kerkis 2020. aasta jooksul ka Eesti internetimaastikule mitmeid uusi portaale ja mobiilirakendusi, mis on loodud aitamaks klientidel leida vajaliku töö sooritamiseks vabakutseline spetsialist ning niisugusel moel elatist teenivatel inimestel jõuda uute klientideni. Üheks selliseks rakenduseks on Tööleidja. Tööleidja idee sündis juristi soovist pakkuda enda teenust vabakutselisena, tehes seda ilma spetsiaalse turunduskampaaniata. Viimaseks ajendiks veebiteenuse loomisel oli eelnevalt mainitud töötuse määra kasv pandeemiast tingitud eriolukorra tõttu Eestis.

Käesoleva bakalaureusetöö eesmärgiks on luua täisfunktsionaalne veebipõhine rakendus ettevõttele Tööleidja. Veebisaidi peamiseks otstarbeks on võimaldada vabakutselistel inimestel võimalikult hõlpsalt uusi kliente leida ja enda oskusi rakendades raha teenida. Lisaks on platvormil võimalik lisada harilikke töökuulutusi, mis on suunatud kasutajatele, kes eelistavad vabakutselisele tööle ametliku töö(võtu)lepingu sõlmimist.

Bakalaureusetöö jaotub viieks sisupeatükiks. Esimeses peatükis antakse ülevaade sarnastest Eestis teostatud tarkvaralahendustest. Teises peatükis kirjeldatakse loodava tarkvara arendusnõudeid. Kolmas peatükk annab ülevaate kasutatud tehnoloogiatest. Neljandas peatükis on kirjeldatud rakenduse arhitektuur ja loomise protsess. Viimane peatükk pühendub valminud rakenduse analüüsimisele.

2. Mõisted ja terminid

API – *Application Programming Interface*, rakendustarkvaraliides

CSS – *Cascading Style Sheets*, W3C poolt standardiseeritud märgistuskeel kujunduse loomiseks veebidokumentidele

CSS-in-JS – *Cascading Style Sheets In JavaScript* - C. Chedeau poolt välja töötatud tehnoloogia CSS'i kirjutamiseks programmeerimiskeeles JavaScript [5]

CORS – *Cross-Origin Resource Sharing*, mehhanism, mis võimaldab veebilehel puuduvaid ressursse (näiteks pilte, fonte, skripte, videosid) võtta teisest domeenist

DOM – *Document Object Model*, W3C poolt standardiseeritud platvormist ja keelest sõltumatu liides, mis suhtleb XML ja HTML dokumentidega

ECMAScript – *ECMA-262*, organisatsiooni Ecma International poolt loodud JavaScripti standard

HTML – *Hypertext Markup Language*, W3C poolt standardiseeritud veebidokumendi struktuuri märgistuskeel

HTTP(S) – *Hypertext Transfer Protocol (Secure)*, protokoll teabe edastamiseks arvutivõrkudes

JSON – *JavaScript Object Notation*, programmeerimiskeelest sõltumatu andmevahetusvorming, mis põhineb programmeerimiskeele JavaScript'i alamhulgal

JSX – *JavaScript XML*, ECMAScript'i laiendus, mis lubab kasutada HTML'i ja XML'ile sarnanevat süntaksit JavaScriptis

REST – *Representational State Transfer*, veebiteenuste programmeerimise paradigma, mille arendas välja R. Fielding [6]

SSL – *Secure Sockets Layer*, sertifikaat, mis tagab, et veebileht, mida brauser kuvab, on see, kuhu külastaja jõuda soovis ning et ühendus on turvaline

UI – *User Interface*, kasutajaliides

VDOM – *Virtual DOM*, dokumendiobjektimudeli (DOM) representatsioon, mis hoiab veebirakenduse kasutajaliidese oleku mälus selle dünaamiliseks uuendamiseks

W3C – *World Wide Web Consortium*, veebistandardeid- ja protokolle arendav organisatsioon

XML – *Extensible Markup Language*, W3C poolt standardiseeritud laiendatav üldotstarbeline märgistuskeel

3. Sarnased tarkvaralahendused Eestis

Selles peatükis võetakse vaatluse alla Eestis realiseeritud Tööleidjale sarnase sisuga rakendused.

Alternatiivsete töösuhete vahendamiseks on Eestis alates 2013. aastast tegutsenud platvormitöö portaal GoWorkaBit. Tegemist on veebilehega, mille fookuses on nii ajutised kui pikemaajalised töösuhted, andes kasutajale vabaduse valida endale sobiv töökoormus. Konkreetse suunaga vabakutselistele teenusepakkujatele leidub Eesti internetimaastikul veebisait Vabakutselised, mis on Tööleidjale otseseks konkurendiks. Sarnaseks portaaliks on ka AbiVaja, kuid töö autoril ei õnnestunud seda veebilehte testida, sest kasutajaks registreerimise üritus lõppes veateatega. Pärast esimest koroonaviiruse lainet 2020. aastal lisandusid Tööleidjale veel kolm sarnast portaali. Ilmavalgust nägid ehitus-, remondi-, sanitaartehtnilisi jms. teenuseid pakkuvaid spetsialiste klientidega ühendavad veebirakendus Gewodo ja mobiilirakendus Handies, ning samuti äpina loodud iseseisvatele spetsialistidele suunatud Skillus.

Järgnevalt on toodud eelnimetatud rakenduste võrdlus:

Tabel 1. Varasemate Eestis loodud Tööleidjaga sarnaste tarkvaralahenduste võrdlus

Portaal	Tüüp	Kaasaegne kasutajaliides	Teenusepakkuja detailne profiilivaade	Kuulutuse lisamine tasuta	Kiirsuhtlus	Konto sidumine Facebook'iga	Konto sidumine Google'iga
GoWorkaBit	Veeb	✓	✓	✗	Ei rakendu	✓	✓
Vabakutselised	Veeb	✗	✗	✓	✓	✓	✗
AbiVaja	Veeb	✓	✗	✗	✗	✗	✗
Gewodo	Veeb	✓	✓	✓	✗	✗	✗
Handies	Mobiilirakendus	✓	✓	✓	✓	✗	✓
Skillus	Mobiilirakendus	✓	✓	✓	✓	✓	✓

Tabelist 1 nähtub, et kõik võrreldavad tarkvarad erinevad teatud omaduste poolest. Uued platvormid Gewodo, Handies ja Skillus võimaldavad teenuse lisamist tasuta ja teenusepakkujaga parema usalduse saavutamiseks saab tutvuda kasutajate profiilidega, aga Gewodo ei toeta lisaks Facebook'i/Google'i konto kaudu autentimise funktsionaalsuse puudumisele kiirsuhtlust. Kõige rohkem funktsionaalsusi omavad Handies ja Skillus, mis on kättesaadavad ainult mobiilirakendusena.

4. Arendusnõuded

Järgnevalt kirjeldab töö autor kliendi esitatud funktsionaalseid ja mittefunktsionaalseid nõudeid loodavale tarkvarale.

4.1 Funktsionaalsed nõuded

Tööleidja sihtrühma kuuluvad vabakutselised spetsialistid, kes soovivad enda teenust reklaamida, kliendid, kelle sooviks on leida abiline vajaliku ülesande sooritamiseks, ning samuti tööd otsivad isikud. Selleks, et veebirakendus toimiks sihipäraselt, peavad olema täidetud järgnevad funktsionaalsed nõuded:

- **teenuse reklaami lisamine ja muutmine**
- **tööpakkumise reklaami lisamine ja muutmine**

Teenuse/tööpakkumise lisamine ja muutmine käivad läbi ankeedi ja võimalik on lisada peale tekstiliste andmete ka pilte.

- **kataloogivaade**
- **üksiku kuulutuse vaade**
- **profiilivaade**

Nii teenuste kui tööpakkumistega tutvumiseks on vaja kataloogivaadet, mis võimaldab:

- kuulutusi filtreerida asukoha ja kategooria alusel,
- kuulutusi sorteerida populaarsuse, kuulutuse lisamise aja ja hinna järgi
- üksiku kuulutuse või kuulutuse looja profiiliga tutvuda

- **kasutajakonto**

Kasutaja peab saama registreerida Tööleidjasse konto ja sellega sisse logida. Autentimine peab olema võimalik ka Google'i või Facebook'i konto kaudu. Kasutajal peab olema võimalik täiendada profiili ja muuta konto sätteid.

- **sõnumite saatmine**

Teenusepakkujale päringu tegemiseks või tööpakkumisele kandideerimiseks on tarvis sõnumite saatmise võimalust vahetuks suhtluseks.

- **informeeriv jalus**

Veebilehel peab olema jalus, mis sisaldab informatsiooni Tööleidja kohta ja linke privaatsuspoliitikale, üldtingimustele ja teistele informeerivatele lehekülgedele.

4.2 Mittefunktsionaalsed nõuded

Mittefunktsionaalsete nõuete alla kuuluvad kriteeriumid, mille alusel hinnatakse süsteemi töökäiku ja kvaliteeti. Üheks olulisemaks mittefunktsionaalseks nõudeks on turvalisus, s.t, et veebileht peab kasutama HTTPS protokollu ja omama kehtivat SSL sertifikaati. Parema nähtavuse saavutamiseks otsingumootorites peab veebisait olema vastavalt optimeeritud, milleks kasutatakse lehe sisu arusaadavuse parandamist otsingumootoritele. See on tuntud kui SEO (*Search Engine Optimization*). Rakendust külastavate kasutajate liiklusvoo jälgimiseks ja analüüsimiseks soovib klient liidestust Google Analytics analüütikatarkvaraga. Viimaseks mittefunktsionaalseks nõudeks on liidestus Facebook Pixel'iga, mis võimaldab ühenduse tarkvaradega Facebook Ads Manager ja Facebook Analytics. Need liidesed aitavad mõõta Facebooki reklaamide efektiivsust ja edendada sihtturundust Facebook'i platvormil.

5. Kasutatud tehnoloogiad

Käesolevas peatükis on välja toodud veebirakenduse loomiseks valitud tehnoloogilised tööriistad. Valikute tegemisel võeti arvesse töövahendite kaasaegsust, kasutajaskonna suurust ja töö autori huvi tehnoloogiatega tutvuda.

5.1 Kliendipoolsed tehnoloogiad

5.1.1 React

React sai alguse 2011. aastal ühismeediaplatvormi Facebook vajadusest efektiivsemalt hallata ja optimeerida muutuseid veebirakenduse kasutajaliideses [7]. Tegemist on nüüdseks kõige populaarsema kliendipoolse JavaScript'i teegiga [8]. React kasutab kasutajaliidese dünaamiliseks renderdamiseks „virtuaalset“ dokumendiobjektimudelit VDOM, mis lubab rakenduses toimuvate muudatuste korral uuesti renderdada ainult neid elemente, mis muutusid. Nagu teisedki nüüdisaegsed *front end* raamistikud, põhineb React komponentidel. Komponent on isoleeritud tükike rakendusest, mis on taaskasutatav erinevates rakenduse osades. Komponent saab React'i programmides olla JavaScript'i klass või funktsioon, mis tagastab HTML'i või JSX'i. JSX ehk JavaScript XML on React'i tarbeks loodud JavaScript'i laiendus, mis on nüüdseks kasutusel ka teistes tehnoloogiates. JSX lubab komponentides defineerida XML'ile sarnanevaid kohandatud elemente ning kasutada JavaScript'i keele võimalusi elementide haldamiseks ja kuvamiseks. Komponentide vahel on võimalik jagada sisendatribuute (*props*) ja komponentide sisesed muutujad saavad omada olekut (*state*). Olekuhaldus (*state management*) on muutujate olekute haldamine reaktiivselt vastavalt kasutaja toimingutele. Olekuid hallatakse nii komponendi-põhiselt kui rakenduse-üleselt ehk globaalselt. Globaalseks olekuhalduseks on Tööleldjas kasutusel React'i arendajate poolt välja töötatud Redux, mida käsitletakse lähemalt järgmises rakenduse arhitektuuri tutvustavas peatükis.

5.1.2 Alternatiivsed *front end* raamistikud

Alternatiividena olid kliendipoolsetest raamistikest kaalutlusel:

- **Vue** – Ligipääsetavusele keskenduv kaasaegne UI raamistik, mis koondab veebilehe struktuuri (HTML), kujunduse (CSS) ja loogika (JavaScript) ühte .vue laiendiga faili. Sarnaselt React'ile kasutab Vue.js VDOM'i ja rajaneb komponentidel, mis jagavad

sisendatribuute ja olekuid. Vue raamistikus on kasutusel HTML'il põhinev *template* süntaks, mis lubab kasutada HTML'i elementidel Vue-spetsiifilisi direktiive. Vue toetab ka JSX'i. Tööleidja portaali loomiseks ei valinud töö autor Vue.js'i, kuna tegemist on võrdlemisi noore ja tagasihoidliku ökosüsteemiga raamistikuga.

- **Svelte** – Valikus olnud *front end* raamistikest kõige uuem, 2016. aastal alguse saanud Svelte ei kasuta „virtuaalset“ DOM'i, vaid kompileerib vahetult komponentide koodi. Selle tulemuseks on puhas JavaScript, mis uuendab dokumendiobjektimudelit. See omadus võimaldab hoida koodibaasi failide suurused minimaalsena ja tagab parema käitusaegse jõudluse [9]. Tegemist on madala õppimiskõveraga unikaalse tehnoloogiaga JavaScript'i raamistike sfääris, kuid ei osutunud käesoleva töö raames valituks ebaküpsuse tõttu.

5.1.3 CSS-in-JS

Veebilehekülgede stiilimiseks loodud keel CSS on mõeldud dokumendi kujundamiseks tervikuna. Sellega kaasneb tänapäevaste mahukate komponentideks jaotatud veebirakenduste puhul stiilide kokkulangemise ja üle kirjutamise oht. Üheks modernseks meetodiks, mis on loodud selle probleemi kõrvaldamiseks, on CSS-in-JS. See tehnoloogia võimaldab stiilide kirjutamiseks kasutada JavaScript'i, mis annab lisaks komponendi-põhisele skoobile juurde dünaamilisi võimalusi kasutajaliidese välimuse muutmiseks. Tööleidjas on veebisaidi kujunduse haldamiseks kasutusel teek styled-components, mis on CSS-in-JS tehnoloogiate hulgas kõige levinum [10]. Lisaks võttis töö autor komponentide kujundamise protsessi kiirendamiseks kasutusse styled-components'i peale ehitatud raamistiku xstyled, millega kaasnevad kasutajaliidese programmeerimist lihtsustavad utiliidid.

5.2 Serveripoolsed tehnoloogiad

5.2.1 Node.js

Tööleidja veebiserveri taga on JavaScript'i käitussüsteem Node.js, mis kasutab JavaScript'i interpreteerimiseks Google'i V8 mootorit. V8 sai algselt loodud ainuüksi Google Chrome'i brauseri jaoks, aga leiab nüüdseks kasutust erinevates tarkvaraprojektides. Node.js võimaldab asünkroonset andmevahetust ja sisaldab HTTP teeki, mis lubab luua ja jooksutada veebiserverit.

Node.js on alustalaks veebimoodulite haldamise süsteemile npm (*Node Package Manager*), millest tuleb lähemalt juttu järgnevas peatükis.

5.2.2 Express

Express.js on pikaajaliselt Node.js ökosüsteemis kasutusel olnud ja nende aastatega standardiks saanud *back end* raamistik, mis teeb Node.js serveri seadistamise lihtsaks ja kiireks. Express'iga kaasneb mugav marsruutimissüsteem kasutajaliidesest HTTP päringute tegemiseks vajalike otspunktide määratlemiseks. Peale selle võimaldab Express lisada serverile *middleware*'e ehk "vahevarasid", mis toimivad vahendajana serveri ja teise teenuse või mooduli vahel, et hõlbustada nende vahelist suhtlust. Vahevarasid kasutatakse näiteks teatud andmevahetusvormingu muundamiseks teise vormingusse või erinevate mehhanismide lubamiseks või keelamiseks serveris (nt. CORS). Tööleldjas on võimalik lisaks tavapärasele meiliaadressi ja parooliga autentimisele kasutajat autentida ka Facebook'i või Google'i konto kaudu. Selleks on serveri poolel vahevarana kasutusel teek Passport.js, mis hõlmab endas üle 500 erineva autentimise võimaluse.

Express'ile oli alternatiivina kaalumise all uus ja modulaarne raamistik Fastify, mis paneb erilist rõhku kiirusele. Käesoleva projekti serveripoolse ülesehitamiseks eelistas töö autor aga Express'i, kuna Fastify on alles Node.js kogukonnas kanda kinnitamas ja Express leiab oluliselt rohkem kasutust [11].

5.2.3 Socket.IO

Sõnumite edastamise reaalajas teeb võimalikuks veebitehnoloogia WebSockets, mis kirjeldab kahepoolset sündmusepõhist suhtlust brauseri ja serveri vahel. Selle võimaldamiseks on Tööleldjas kasutusel teek Socket.IO, mis koosneb serveripoolsest Node.js serverist ja kliendipoolsest JavaScripti teegist socket.io-client, mis suhtleb serveriga.

5.2.4 Andmebaas

Andmete hoiustamiseks on Tööleldja veebirakenduses kasutusel avatud lähtekoodiga mitterelatsiooniline (*NoSQL*) andmebaas MongoDB. Võrreldes traditsioonilise SQL andmebaasiga, mis kasutab andmete säilitamiseks struktureeritud tabeleid, salvestatakse andmed MongoDB's kollektsoonide ja neis sisalduvate BSON-dokumentidena. BSON'i näol on tegemist JSON formaadiga binaarkujul. Dokument koosneb võtme ja väärtuse paaridest, mis

korreleeruvad JavaScript'i objektidega. Andmebaasina on MongoDB skaleeritav, paindlik, kiire ning omab rikkalikult võimalusi päringute tegemiseks. Sellele lisaks on MongoDB andmete mudeldamiseks saadaval Node.js moodul Mongoose, mille abil on hõlbus JavaScript'i failides andmete tüüpe määrata, andmeid valideerida ja päringuid ehitada.

6. Rakenduse arhitektuur ja arendus

Käesolevas peatükis annab töö autor ülevaate tarkvara arhitektuurist, arendusprotsessist ja selle käigus kasutatud töövahenditest.

Nüüdisaegsed JavaScript'il põhinevad veebirakendused on ehitatud Node.js moodulitele, mis kätkevad endas spetsiifilist funktsionaalsust võimaldavat taaskasutatavat JavaScript'i koodi. Igal moodulil on oma kontekst ehk neis sisalduvad väärtused ja avaldised ei ole seoses teiste moodulitega. Seeläbi ei saasta moodulid programmide globaalset skoopi. Moodulite haldamiseks on loodud npm (*Node Package Manager*), mis on nüüdseks kujunenud maailma suurimaks tarkvara registriks [12]. Npm'iga kaasneb mugav käsurealiides, mis lubab mooduleid alla laadida ja installeerida. Tööleidja rakenduses on moodulite haldamiseks kasutusel npm nii kliendi- kui serveripoolel.

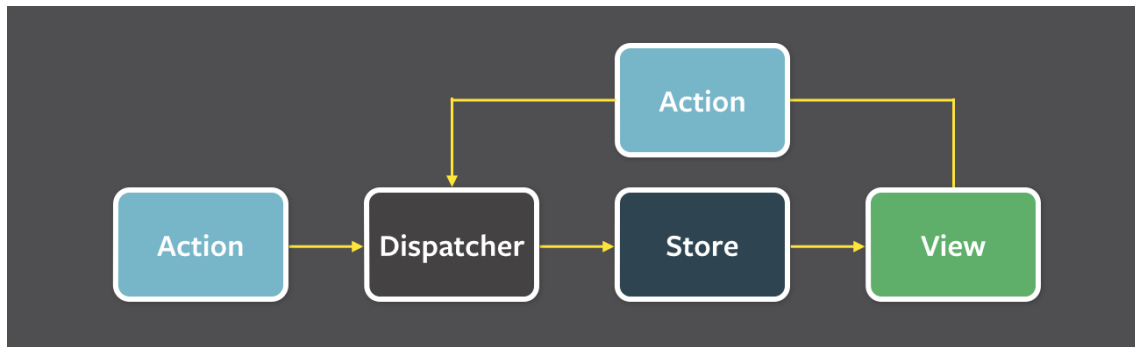
6.1 Kasutajaliides

React'i kasutava vajalike funktsionaalsustega projekti ülesseadmine võib osutuda keerukaks. Protsessi lihtsustamiseks lõi React'i arendajate meeskond käsurealt jooksutatava eelseadistatud vahendi create-react-app, millele on ka Tööleidja *front end* rajatud. Create-react-app toetab React'i, JSX'i ja uusima JavaScript'i süntaksit, mille teeb võimalikuks JavaScript'i kompilaator Babel. Arenduses kasutatavate failide ja moodulite kokku pakkimine ning optimeerimine staatilisteks failideks on võimaldatud Webpack'i poolt. Webpack lubab ka JavaScript'i failidesse importida staatilisi ressursse ja kompileerib jooksvalt koodi, mis hõlbustab arendusprotsessi. Create-react-app vastutab ka koodi puhtuse ja vigade eest, kuvades käsureal või veebilehitsejas vastavaid teateid, mida võimaldab koodi analüüsimise vahend ESLint.

Andmevoo haldamiseks React'i rakendustes leidub erinevaid paradigmasid. Tööleidja projekti tarbeks võttis töö autor kasutusse React'i arendajate poolt populariseeritud arhitektuurimustri Flux. Kõnealune arhitektuur jaotab rakenduse järgnevateks osadeks:

- *Store* ehk andmeladu on JavaScript'i objekt, mis hoiab endas rakenduse-üleseid olekuid ehk andmete väärtuseid.
- *Dispatcher* ehk dispetšer on objekt, mis teavitab andmeladu sündmustest kasutajaliideses.

- *Views* ehk vaated on kasutajaliidese komponendid. Vaated kuuluvad muutuseid laos ja teevad vastavaid muudatusi kuvamises.
- *Actions* ehk toimingud on objektid, mis sisaldavad kogu vajalikku informatsiooni ühe toimingu teostamiseks.



Joonis 1. Andmevoog Flux arhitektuuriga React'i rakenduses [13]

Flux mustri rakendamiseks Tööleldjas on kasutusel teek Redux, mille kasutusele võtmiseks on vajalik rakenduse ülemkomponent mässida Redux'iga kaasnevasse Provider komponenti. Provider teeb Reduxi lao kogu rakendusele kättesaadavaks.

```

import { render } from "react-dom";
import App from "../App";
import { Provider } from "react-redux";
import store from "../store"

const rootElement = document.getElementById("root");

render(
  <Provider store={store}>
    <App />
  </Provider>,
  rootElement
)

```

Joonis 2. Näidis rakenduse ülemkomponendi (App) sidumisest globaalse olekuhaldusega

Järgnevalt on toodud näide sõnumite saatmise funktsionaalsuse raames hallatavatest olekutest ja nendega seotud toimingute defineerimisest. Reduktor (*reducer*) koondab Flux arhitektuuris enda alla toimingute kirjeldused. Objektis *initialState* defineeritakse muutujate algseid olekud, mida hakatakse rakenduses toimingute abil muutma.

```

const initialState = {
  chats: {},
  onlineUsers: {},
  unreadCount: 0,
};

const reducer = (state = initialState, action) => {
  switch (action.type) {
    case "SET_CHATS":
      return {
        ...state,
        chats: action.payload,
      };
    case "SET_ONLINE_USERS":
      return {
        ...state,
        onlineUsers: action.payload,
      }
    case "SET_UNREAD_COUNT":
      return {
        ...state,
        unreadCount: action.payload,
      }
    default:
      return state;
  }
};

export default reducer;

```

Joonis 3. Näidis Redux'i toiminguid koondavast reduktorist

Soovitud toimingu läbiviimiseks saab valitud komponendis välja kutsuda dispetšeri, mis annab laole märku, et toimus muudatus:

```

dispatch({ type: "SET_CHATS", payload: conversations });

```

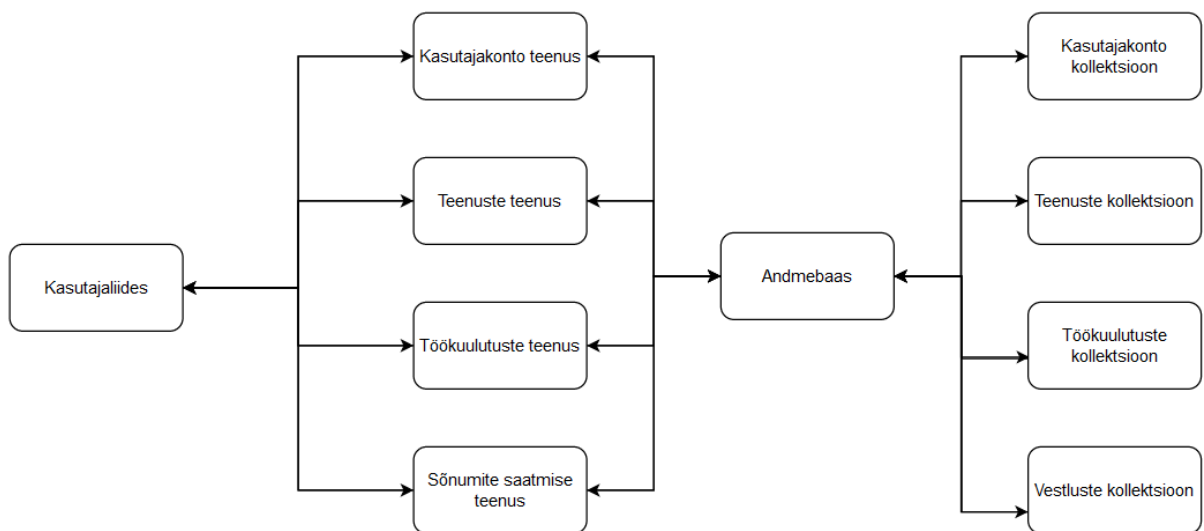
Joonis 4. Näidis Redux'i dispetšerist

6.2 Server ja andmebaas

Tööleidja veebirakendus on üles ehitatud kasutamaks REST tarkvaraarhitektuuri laadi HTTP päringute tegemiseks. Vastuse päringule saadab server kliendile JSON vormingus. Vastus sisaldab staatuse koodi, mis annab teada, kas tehtud päring oli edukas või esines probleem. Olenevalt päringust sisaldab vastus ka andmeid, mida küsiti. Levinumate toimingute hulka

kuuluvad GET andmehulga pärimiseks serverilt, POST andmete edastamiseks serverile, PUT andmete asendamiseks serveris ja DELETE soovitud andmete kustutamiseks serverist.

Käsitletavas veebirakenduses on *back end* jaotatud mikroteenusteks. S.t, et igale rakenduse põhifunktsionaalsusele eksisteerib eraldi serveripoolne koodibaas. Mikroteenuste arhitektuuril põhinevat tarkvara on lihtsam mõista, arendada ja testida, kui monoliitset rakendust [14]. Joonisel 5 on välja toodud diagramm Tööleidja rakenduse arhitektuurist.



Joonis 5. Tarkvaraarhitektuur Tööleidja rakenduses

Joonisel 6 on kujutatud näide JavaScript'is defineeritud töökuulutuse MongoDB dokumendi skeemist (lihtsustatud):

```

const jobSchema = mongoose.Schema(
  {
    status: {
      type: Boolean,
      default: true,
    },
    category: {
      type: String,
      default: null,
    },
    position: {
      type: String,
      default: null,
    },
    name: {
      type: String,
      default: null,
    },
    description: {
      type: String,
      default: null,
    },
    expectations: {
      type: String,
      default: null,
    },
    validUntil: {
      type: Date,
      default: null,
    },
    images: [
      {
        _id: false,
        small: {
          type: String,
        },
        medium: {
          type: String,
        },
        large: {
          type: String,
        },
      },
    ],
  }
);

```

Joonis 6. Näidis MongoDB dokumendi skeemist

Järgnevalt toob töö autor välja mõningad näited päringutest, mida on võimalik töökuulutuse andmestikule tarnitud rakenduses teha:

Tabel 2. Näidis võimalikest päringutest töökuulutusi haldavale mikroteenusele

Meetod	Päring	Päringu keha/parameetrid	Tulemus
GET	/job	params: { „id“: „235235fe34“ }	Töökuulutuste kollektsiooni üksiku kuulutuse dokument vastava ID'ga
GET	/jobs	params: { „location“: „Võru“, „category“: „Fotograafia“, „date“: "2020-01-23T00:00:00.441Z", page: 1 }	Objekt kõigist töökuulutustest, mis vastavad päritavatele parameetritele
POST	/job	body: { formData: formData }	Uus töökuulutuse dokument lisatakse kollektsiooni
DELETE	/job	params: { „id“: „711223ae52“ }, body: { „userId“: „3334990e00“ },	Kollektsioonist eemaldatakse vastava ID'ga dokument

Selleks, et MongoDB dokumentide skeeme ei peaks igas mikroteenuses uuesti defineerima ja muudatuste korral igas mikroteenuses eraldi muudatusi tegema, võttis töö autor kasutusele Git'i

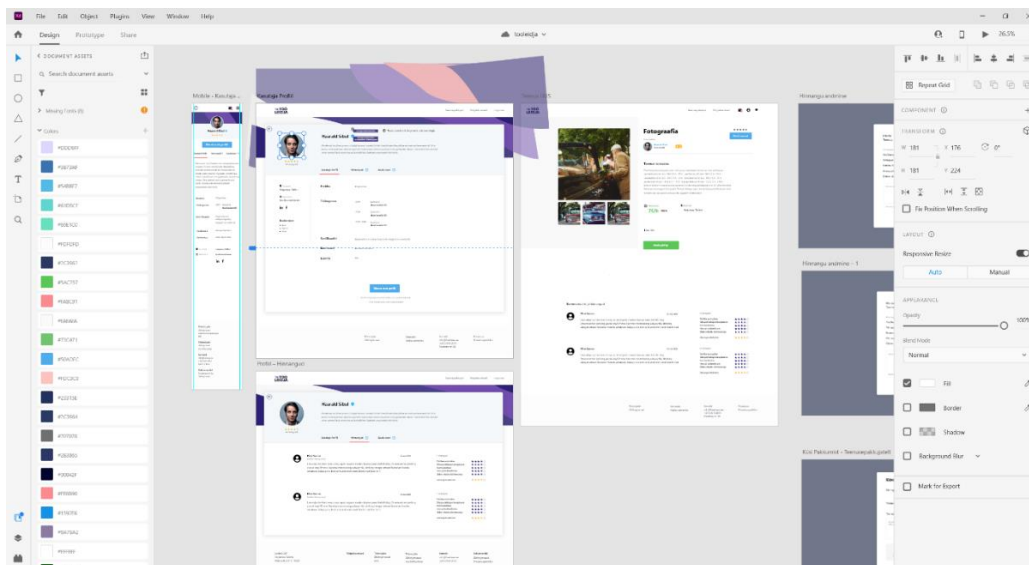
alammoodulid (*submodules*). S.t, et skeemid on hallatavad Git'i repositooriumina, mida on võimalik lisada alamprogrammina iga mikroteenuse juurde:

```
git submodule add https://github.com/kasutajanimi/tooleidja-models.git models
```

Joonis 7. Näidis Git'i alammooduli sidumisest projektiga

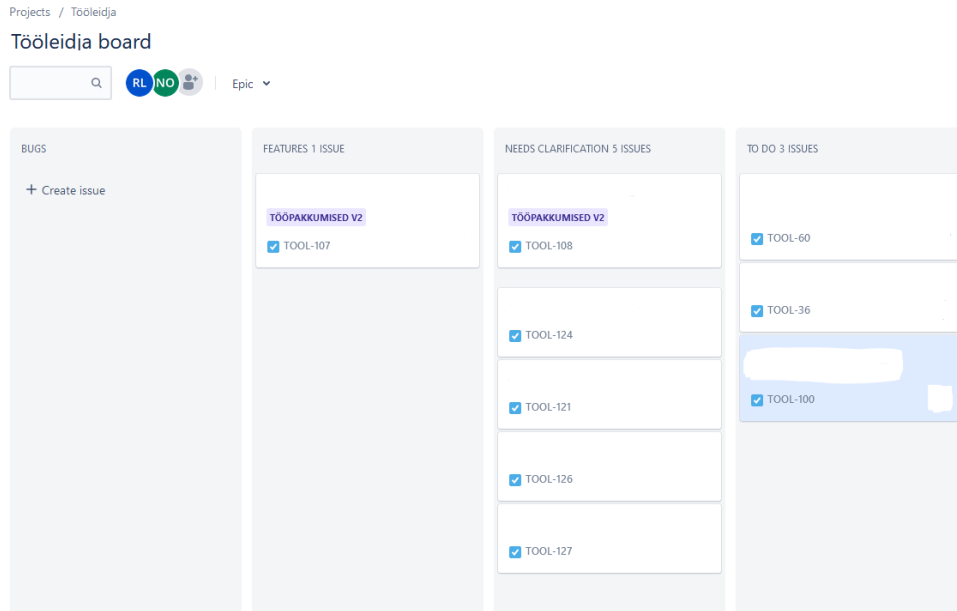
6.3 Töövahendid

- **Visual Studio Code** – Microsoft'i poolt loodud avaliku lähtekoodiga tekstiredaktor, mis on rajatud Electron'i tehnoloogiale. Electron on platvormist sõltumatute programmide kirjutamiseks mõeldud raamistik, mis kasutab väljaspool veebi ökosüsteemi tarkvara loomiseks HTML'i, CSS'i ja JavaScript'i. Töö autor valis Visual Studio Code'i veebirakenduse kirja panemiseks selle laia valiku tööprotsessi lihtsustavate laienduste, versioonihalduse (Git) toe ja kasutusmugavuse tõttu (kiire ja vajadustele kohaldatav).
- **Adobe XD** – Platvormi loomisel toimus koostöö UI disaineriga, kelle disainide põhjal töö autor kasutajaliidese koostas. Kujundused loodi prototüüpimiskeskkonnas Adobe XD. Kujundusfailidest oli töö autoril võimalik välja võtta PNG-failidena pilte ning SVG-failidena illustratsioone ja ikoone.



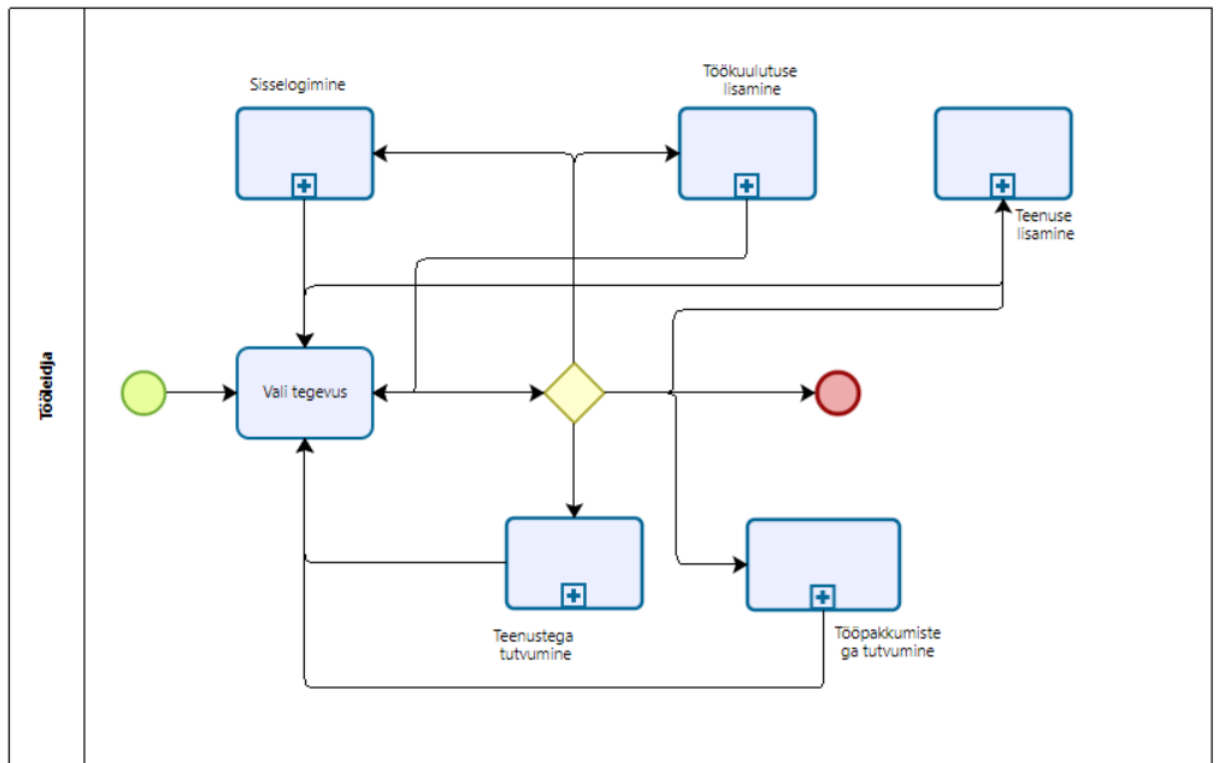
Joonis 8. Adobe XD kasutajaliides

- **Jira** – Projektihaldusele suunatud tarkvara, mis põhineb Kanban metoodikal. Tarkvaraarenduses on selle metoodika kohaselt levinud projekti jaotamine nimekirjadeks, mis koosnevad ülesannetest. Nimekirjad on enamasti kuvatud tahvlina üksteise kõrval, vastavalt ülesannete tüübile või staatusele. Tööleidja kontekstis olid staatusteks näiteks “Vead”, “Uued funktsionaalsused”, “Ootel”, “Arenduses” jne.



Joonis 9. Jira tahvel (osaline)

- **Git** – Koodis tehtud muudatuste jälgimiseks ja haldamiseks võttis töö autor appi versioonihaldustarkvara Git ja majutas Tööleidja repositooriumid GitHub'i platvormile. GitHub annab statistilise ülevaate projektist, võimaldab koodi dokumenteerida ja näha detailselt kõiki muudatusi, mis repositooriumisse üles laetakse.
- **Bizagi Modeler** – Priivarana kättesaadav protsesside modelleerimise ja simuleerimise tööriist, mis annab ülevaate kasutajavoogudest kasutajaliideses ja on abiks rakenduse voogude puuduste leidmisel ja optimeerimisel. Joonisel 10 on kujutatud Tööleidja avalehe protsessimudel.



Joonis 10. Tööleidja avalehe Bizagi mudel

Alamprotsesside diagrammid on kaasatud käesoleva töö lisadesse.

7. Tulemuse ülevaade ja analüüs

Selles peatükis vaadeldakse valminud veebilehe struktuuri ja funktsionaalsusi. Rakenduse testijateks olid peale töö autori klient ja valitud vabakutselised spetsialistid.

Potentsiaalsete lõppkasutajate peal testimise käigus ilmnisid erisugused programmivead. Leidus eksimusi, mida arenduse käigus sooritatud töö autori testid ei tuvastanud. Seda peamiselt seetõttu, et arenduskeskkonnas läbiti teekonnad rakenduses ühekülselt, mis ei ühtinud täpselt testkasutajate toimingutega. Lisaks talitusloogikakihi avastatud vigadele selgusid erinevatele veebilehitsejatele, seadmetele ja nende ekraanisuurustele kohandamisest tingitud visuaalsed deviatsioonid. Kõige rohkem kujunduslikke ebakõlasid esines mobiiltelefoni iPhone ja selle tootja Apple'i brauseris Safari, kuna töö autoril puudus Apple'i seadmetel testide läbiviimise võimalus. Vastavalt tagasisidele tehti korrekture ja korraldati testimist iteratsioonidena, kuni kõik vead olid kõrvaldatud ja kasutajakogemus sujuv. Rakendus on kättesaadav domeenilt www.tooleidja.ee.

Tarnitud veebirakenduse päises paiknevad Tööleidja logo ja lingid, mis viivad:

- teenuste kuulutuste kataloogivaatesse,
- töökuulutuste kataloogivaatesse,
- registreerima või sisse logima.



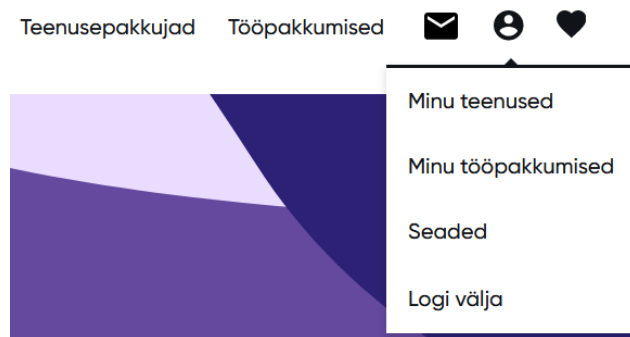
Joonis 11. Tööleidja logo päises

Teenusepakkujad Tööpakkumised Registreeri Logi sisse

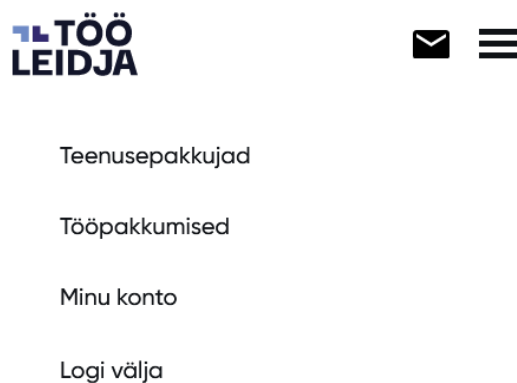
Joonis 12. Lingid algses päises

Kui kasutaja on sisse loginud, tekivad „Registreeri“ ja „Logi sisse“ linkide asemele postkasti ikoon navigeerimaks sõnumite vahetamise lehele, kasutajakonto ikoon koos alammenüüga, mida kuvatakse hiirega kasutajakontole liikudes, ja südametooniga ikoon lemmikuks lisatud teenuste lehele

navigeerimiseks. Kasutajakonto alammenüü koosneb linkidest kasutaja lisatud teenustele, kasutaja lisatud töökuulutustele, kontoseadetele ja välja logimisele.

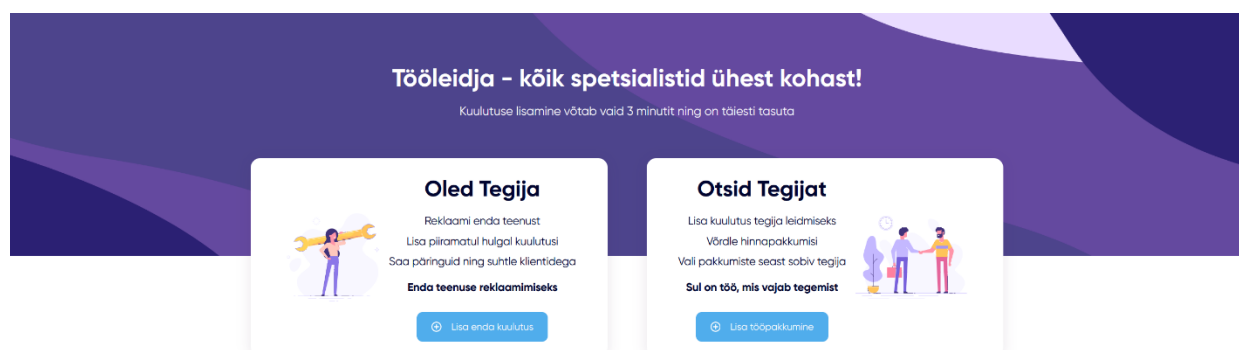


Joonis 13. Navigatsioonimenüü vaade sisse logitud kasutajale



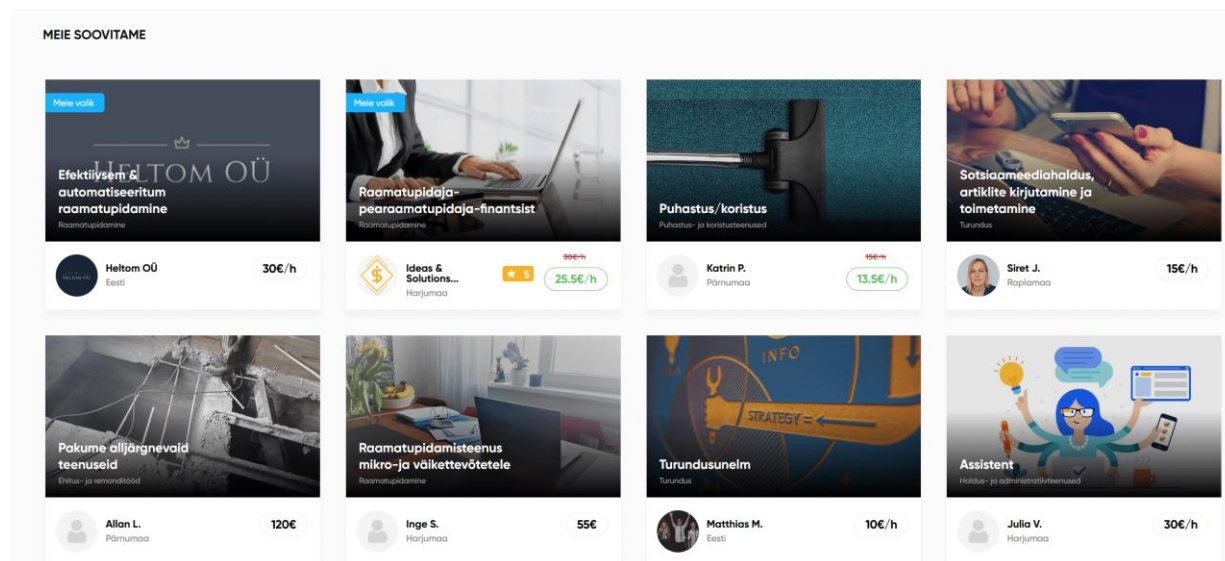
Joonis 14. Navigatsioonimenüü sisse logitud kasutajale mobiilis

Päise all on kuvatud lakoonilise informatsiooniga Tööleidja kohta ja *call-to-action* nuppudega jaotis. Nupud võimaldavad alustada kas teenuse kuulutuse lisamise või töökuulutuse lisamise alustamist.



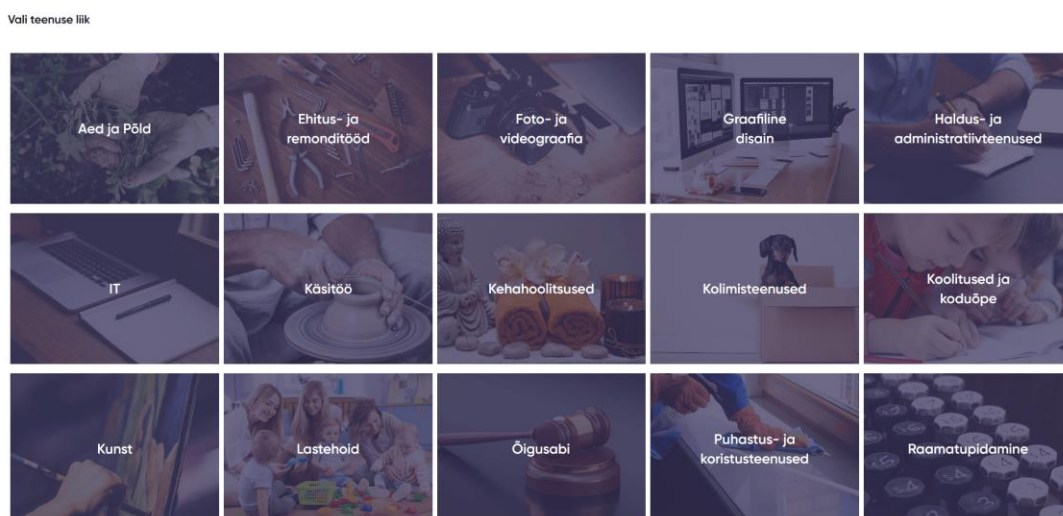
Joonis 15. Reklaami jaotis

Seejärel avaneb ülevaade kliendi poolt esilehel kuvamiseks valitud teenustele. Üksiku teenuse puhul on välja toodud pealkiri, kategooria, teenuse looja, hinnang, asukoht ja hind. Võimalik on navigeerida üksiku teenusega või teenuse looja profiiliga tutvuma.



Joonis 16. Soovitatud teenuste jaotis (osaline)

Eelnevale osale järgneb ülevaade veebilehel pakutavate teenuste kategooriatest ja võimalik on kategooriale klikkida, et filtreerida loodud kuulutusi valitud kategooria alusel.



Joonis 17. Ülevaade kategooriatest (osaline)

Viimaks on avalehel jalus Tööleidja kontaktandmete ja linkidega olulistele lehekülgedele nagu „Üldtingimused“, „Korduma kippuvad küsimused“ ja „Privaatsuspoliitika“.

Leiatöö OÜ Küütri 3, 51007 Tartu, Eesti	Töötajale Üldtingimused KKK	Tööandjale Üldtingimused Lisa töökuulutus	Kontakt info@tooleidja.ee +372 5345 3841	Dokumendid Üldtingimused Privaatsuspoliitika
---	-----------------------------------	---	--	--

Joonis 18. Jalus

Vajutades päise all paikneva osa juures nupule „Lisa enda kuulutus“ sooviga enda teenust reklaamida, avaneb ankeet selle tegevuse teostamiseks. Võimalik on valida teenuse kategooria, pealkiri, asukoht, lühitutvustus, hind, soovi korral soodushind ja lisada pilt/pildid.

Lisa teenus

Teenuse üldinfo
Teenuse liik*
Vali kategooria ↓
Teenuse nimi*
Sisesta teenuse nimi
Maakond*
Vali maakond ↓
Linn/Küla
Sisesta linn/küla
Teenuse tutvustus*
B I
Sisesta pakutava teenuse lühikirjeldus ning ennast või enda ettevõtet tutvustav tekst

Joonis 19. Teenuse reklaami lisamise ankeet (osaline)

Samamoodi, vajutades nupule „Lisa tööpakkumine“, avaneb ankeet selle teostamiseks. Võimalik on valida tööpakkumise kategooria, ametikoht, pealkiri, lühitutvustus, ootused kandidaadile, hüved, asukoht, netopalk, kuulutuse kehtivusaeg ja lisada pilt/pildid.

Lisa tööpakkumine

Tööpakkumise informatsioon

Tööpakkumise kategooria*

Vali kategooria



Ei leidnud sobivat kategooriat?

Vali "VARIA" ning kirjuta, millist kategooriat vajad. Me lisame selle sobivusel juurde.

Ametikoht*

Sisesta ametikoha nimetus

max 20 tähtemärki

Tööpakkumise nimi*

Lisa kuulutusele pealkiri

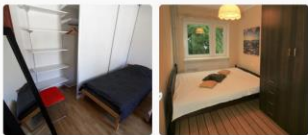

Tööpakkumise tutvustus*

B I

Sisesta pakutava tööpakkumise lühikirjeldus ning ennast või enda ettevõtet tutvustav tekst

Joonis 20. Töökuulutuse reklaami lisamise ankeet (osaline)

Navigeerides üksikule teenusele, on võimalik näha ankeedi kaudu lisatud teenuse detaile, hinnata teenust ja saata teenusepakkujale päring.




Koristus- ja haldusteenused (kodu, kontor, kinnistu)

★★★★★

Hinda teenust

Teenuse pakkuja:

Kevin K.
Vaata profilli

Teenuse tutvustus

Pakume kinnisvara haldust ning koristusteenuseid. Meie peamiseks töövaldkondadeks on kontorite koristamine ja pindade desinfitseerimine, kodukoristus, kinnistute haldus ja välkoristustööd. Lisaks haldame ka külaliskortereid. Kui vajate abi mõne töö osas, andke julgelt endast märku! Võtke ühendust ka juhul, kui töö on nõ. "ebastandardne". Katsume leida parima lahenduse! Võimalik nii ühekordne teenuse osutamine kui ka sobivusel pikaajalisem koostöö.

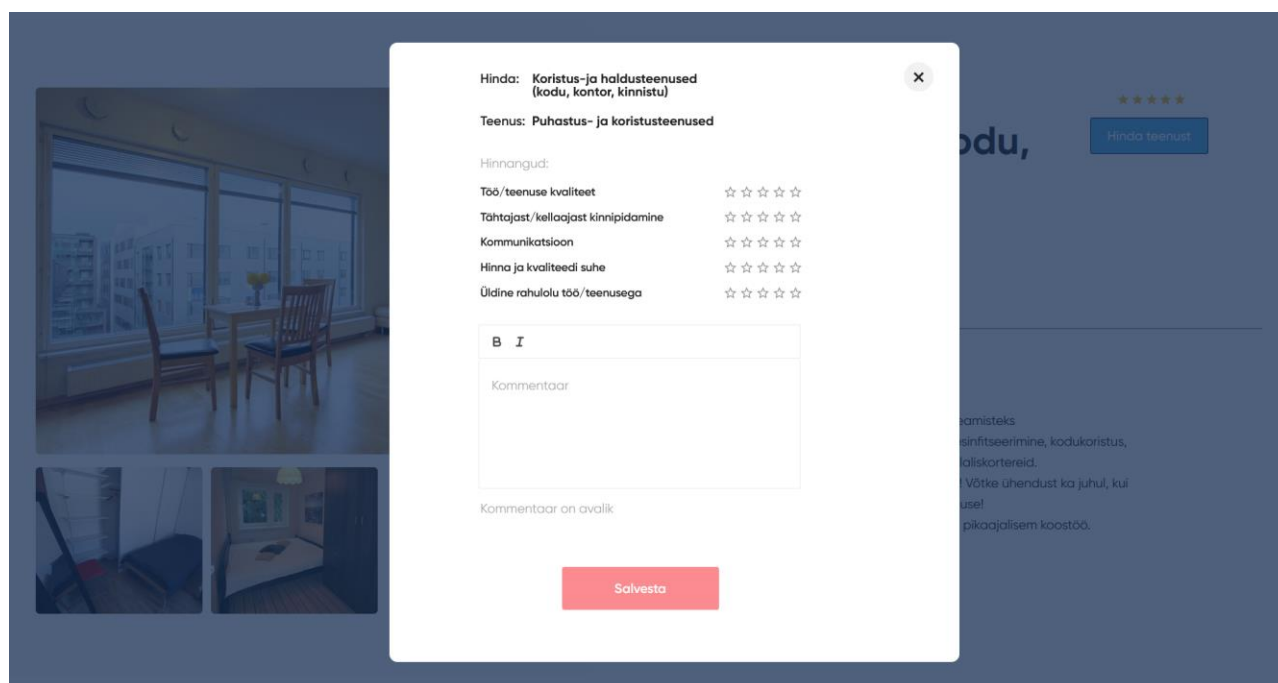
Maksumus

19.8€/h 22€/h

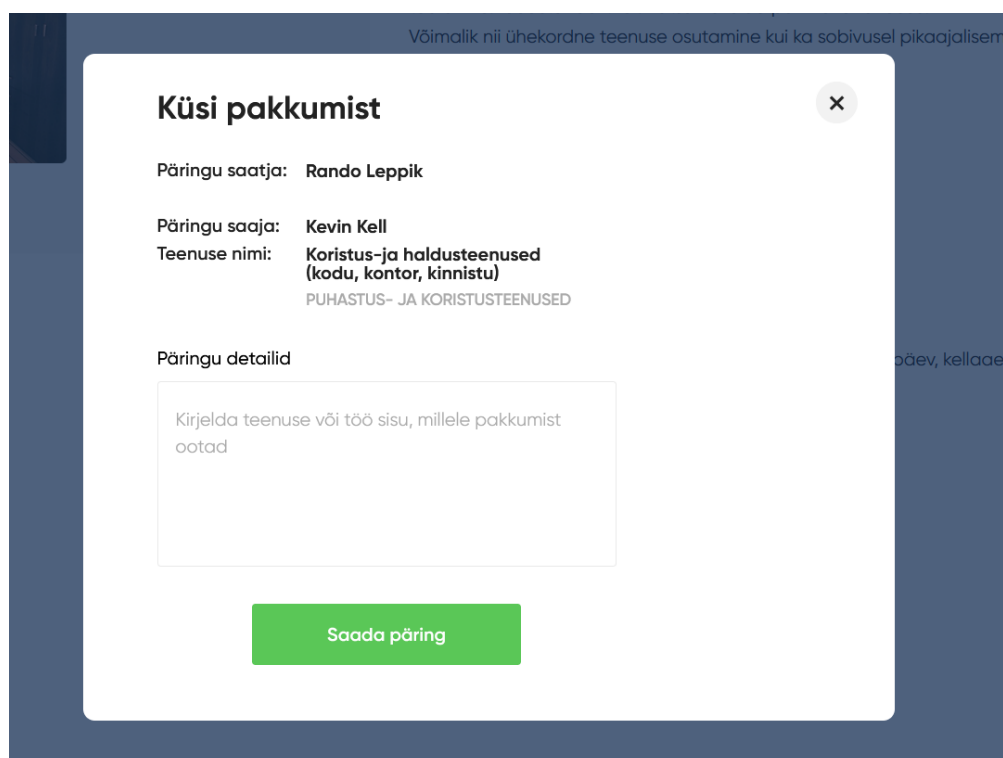
Asukoht

Harjumaa, Tallinn

Joonis 21. Osa üksiku teenuse vaatest

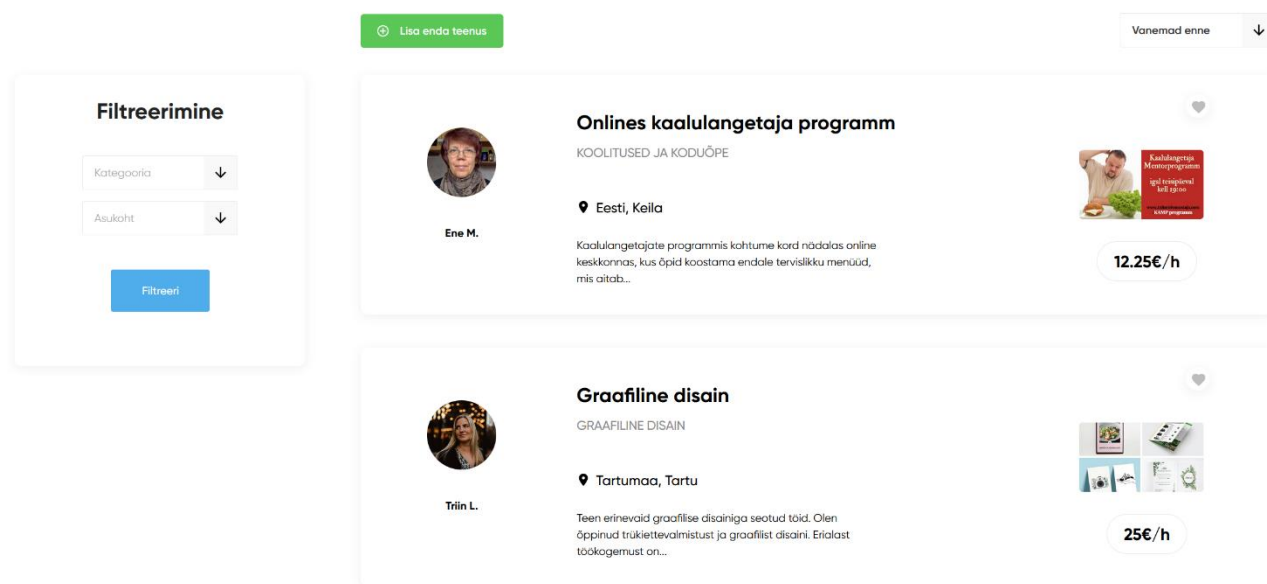


Joonis 22. Teenuse hindamise dialoogiaken



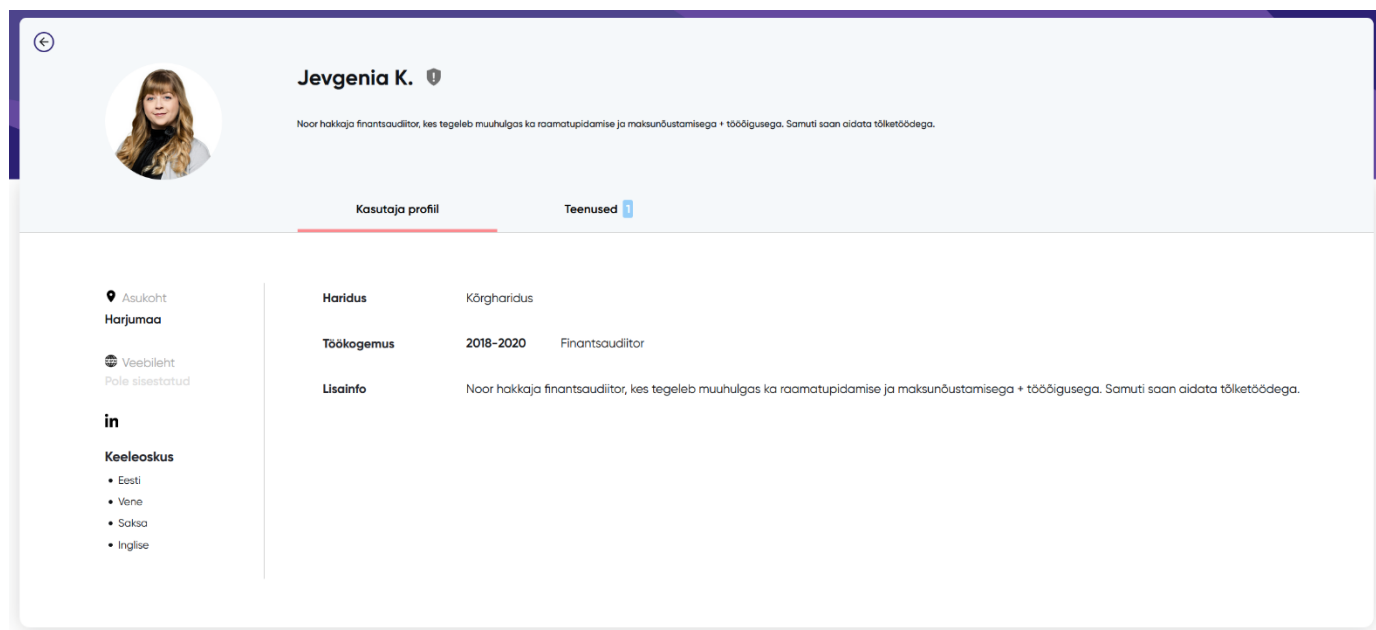
Joonis 23. Dialoogiaken teenusele päringu saatmiseks

Nii teenusepakkujate, kui tööpakkumiste linkidele vajutamine navigatsioonimenüüs viivad ühesuguse kujundusega kataloogivaatesse. Selles vaates saab kuulutusi filtreerida kategooria ja/või asukoha alusel. Lisaks saab kuulutuste järjekorda sorteerida vastavalt populaarsusele, sooduspakkumistele, lisamise ajale ja kuulutuse hinnale. Kataloogivaade lubab navigeerida üksiku kuulutuse või kuulutuse looja profiili vaatesse.



Joonis 24. Teenuste kataloogivaade (osaline)

Teenuse/tööpakkumise looja profiilil on kuvatud vastavalt kasutaja sisestatud andmetele ees- ja perekonnanimi, lühitutvustus, lisainfo, asukoht, veebilehe link, sotsiaalmeedia lingid (Facebook ja Instagram), keeleoskus, haridustase, töökogemus, omandatud sertifikaadid ja läbitud koolitused. Lisaks on võimalik näha kasutaja lisatud teenuseid (kui neid eksisteerib) ja kasutajale antud hinnanguid (kui neid on).



Joonis 25. Tööleidja profiilivaade

Liikudes kasutajakonto vaatesse, avaneb vaade kasutaja enda profiilile, mille kaudu on võimalik ka profiili täiendada. Konto vaates on kättesaadav kõrvalmenüü, milles sisalduvad lingid „Minu profiil“, „Minu sõnumid“, „Pakun teenust“, „Pakun tööd“, „Lemmikud“ ja „Seaded“. Samuti näeb kasutaja enda profiili täituvusprotsenti.



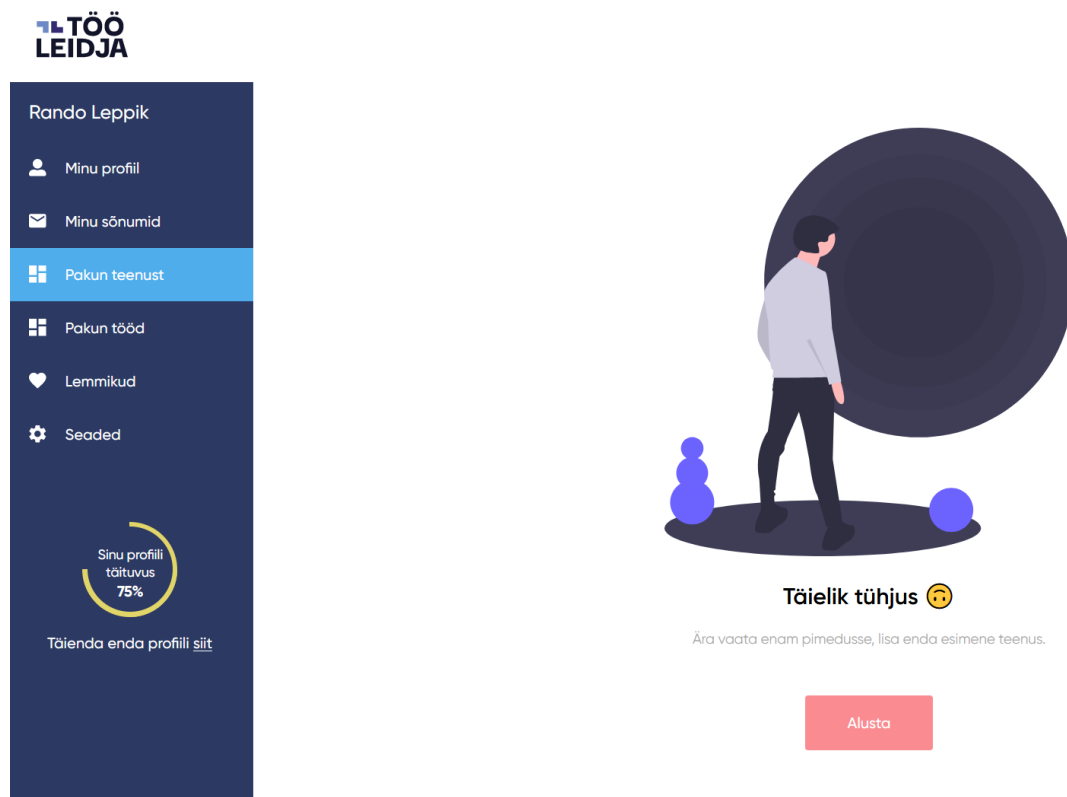
Joonis 26. Kõrvalmenüü kasutajakonto vaates

Seadete alt on võimalik kasutajal muuta ees -ja perekonnanime, lisada/muuta telefoninumbrit, muuta meiliaadressit, kuhu päringut suunata, muuta salasõna ja ettevõtte korral muuta konto liiki ning sisestada ettevõtte andmeid.

Joonis 27. Konto seaded

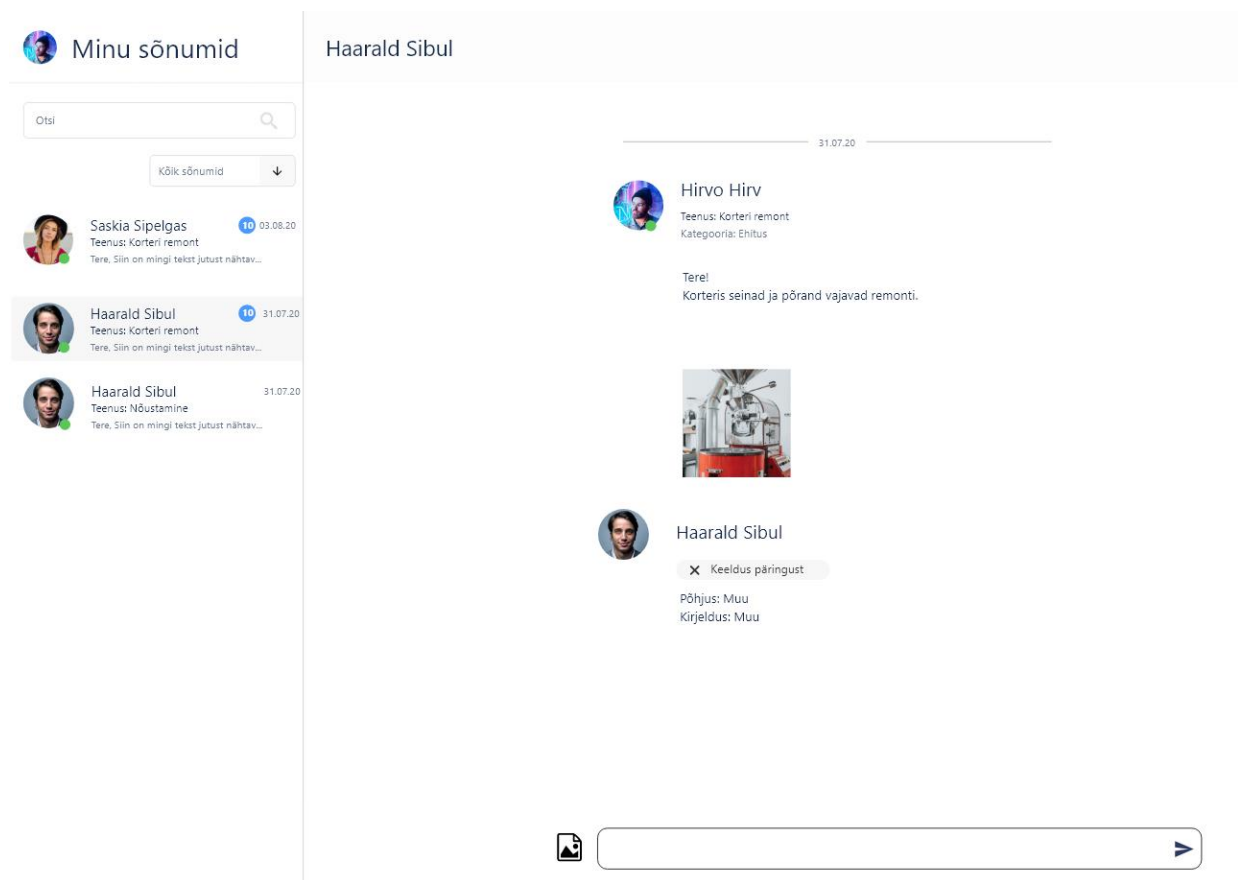
Link „Lemmikud“ viib ülevaatele lemmikuks lisatud kuulutustest ja lingid „Pakun tööd“ ning „Pakun teenust“ koondavad kasutaja poolt lisatud kuulutused tabelina ja annavad võimaluse kuulutust muuta või kuulutus kustutada.

Kasutaja teenuste vaade juhul, kui ühtegi kuulutust ei ole lisatud:



Joonis 28. Kasutajakonto teenuste vaade alguses olekus

Navigeerides konto vaates „Minu sõnumitele“, näeb kasutaja talle saadetud või kasutaja loodud päringuid, saab vestlusi otsida vestluspartneri nime järgi ning filtreerida aja ja staatuse (loetud või lugemata) järgi. Kasutajal on võimalik avada üksik vestlus ja saata sõnumeid reaalsajas. Vestlusaknas on kuvatud vestluse osaliste nimed, profiilipildid, olekud (*online* või *offline*) ja saadetud sõnumite kellaajad. Võimalik on saata ka pilte.



Joonis 29. Vaade „Minu sõnumid“

Ehkki kõik üles seatud funktsionaalsed nõuded said täidetud, ei ole rakendus lõplik. Lisafunktsionaalsusena on kaalutlusel maksesüsteem, mis võimaldaks kliendil tasuda teenusepakujale otse läbi Tööleidja platvormi. Lisaks plaanib töö autor koodibaasi optimaalsemaks administreerimiseks JavaScript'i välja vahetada TypeScript'i vastu. TypeScript lubab lisada programmisestele muutujatele tüüpe. See on loodud aitamaks veebiarendajatel vältida vigu, kirjeldades täpselt missuguseid ja mis tüüpi (sõne, number jne.) muutujaid spetsiifiline komponent või andmestruktuur kasutab. Ühtlasi on plaanis Tööleidjale luua kujundussüsteem (*design system*). See võimaldaks fikseerida ja dokumenteerida erinevad kujunduselemendid ning nendega seotud suurused, värvid ja muud atribuudid. Niisuguste taaskasutatavate elementide olemasolu vähendaks üleliigse CSS'i kirjutamist ja kiirendaks kasutajaliidese programmeerimise protsessi. Tulevikku vaadates on viimaseks ideeks Tööleidja mobiilirakendus mugavaks kasutuseks väljaspool veebilehitsejat.

8. Kokkuvõte

Käesoleva bakalaureusetöö eesmärgiks oli luua toimiv veebirakendus vabakutselistele spetsialistidele ja nende potentsiaalsetele klientidele. Järgnevas lõigus annab töö autor ülevaate, mil määral vastab loodud rakendus püstitatud funktsionaalsetele nõuetele.

Valminud rakenduses on teostatud nii teenuse kui tööpakkumise reklaami lisamise ja muutmise funktsionaalsused. Kasutaja saab reklaami juurde lisada ka pilte. Mõlemat tüüpi kuulutustega on võimalik tutvuda kataloogivaate kaudu, mille kasutajaliides lubab kuulutusi filtreerida ja sorteerida. Samuti saab selle vaate kaudu navigeerida spetsiifilise kuulutuse või kuulutuse lisaja profiili leheküljele. Tööleidja eesmärgipäraseks kasutamiseks on võimalik luua meiliaadressil põhinev isiklik konto, mis on kaitstud salasõnaga. Kasutajakonto lubab muuta sätteid nagu näiteks ees- ja perekonnanimi, e-maili aadress ja telefoninumber. Samuti saab lisada ja täiendada avalikul profiilil kuvatavaid andmeid nagu töökogemus, haridus jms. Kasutajad saavad portaali kaudu üksteisele sõnumeid ja pilte saata. Veebilehel on ka kontaktandmete ja muu kasutajale olulise informatsiooniga jalus.

Tööleidja platvormi tulevikuplaanide hulka kuuluvad maksesüsteem portaali kaudu teenuste eest tasumise jaoks, JavaScript'i välja vahetamine TypeScript'i vastu kasvava koodibaasi paremaks haldamiseks, disainisüsteemi loomine üleliigse CSS'i kirjutamise vältimiseks ja mobiiliäpp, et rakendust oleks võimalik kasutada peale veebilehitseja ka mobiiliseadmes.

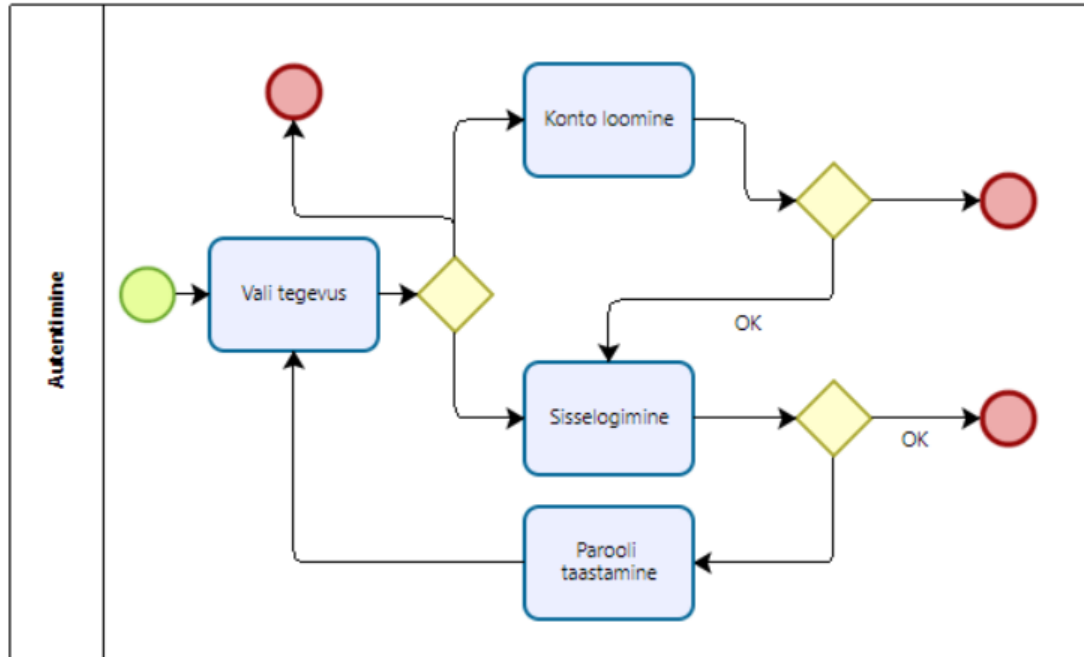
9. Viidatud kirjandus

- [1] International Labour Organization Monitor: COVID-19 and the world of work. Seventh edition, 2021.
https://www.ilo.org/wcmsp5/groups/public/@dgreports/@dcomm/documents/briefingnote/wcms_767028.pdf (20.03.2021)
- [2] Seppel A. Koroonaeptideemia taganedes astub esile töötus, 2020.
<https://majandus24.postimees.ee/6982801/koroonaeptideemia-taganedes-astub-esile-tootus> (20.03.2021)
- [3] Berliner U. Jobs in the pandemic: More are freelance and may stay that way forever, 2020.
<https://www.npr.org/2020/09/16/912744566/jobs-in-the-pandemic-more-are-freelance-and-may-stay-that-way-forever> (21.03.2021)
- [4] Raag T. Uuring: 8 protsenti eestlastest teevad platvormitööd, 2019.
<https://www.pealinn.ee/koik-uudised/uuring-8-protsenti-eestlastest-teevad-platvormitood-n244057> (21.03.2021)
- [5] Ward D. A brief history of CSS-in-JS: How we got here and where we're going, 2017.
<https://levelup.gitconnected.com/a-brief-history-of-css-in-js-how-we-got-here-and-where-were-going-ea6261c19f04> (06.05.2021)
- [6] <https://akit.cyber.ee/term/10843%20AKIT%20REST> (21.04.2021)
- [7] Shaleynikov A. The Evolution of React, 2018. <https://hackernoon.com/the-evolution-of-react-48409fac2efd> (26.04.2021)
- [8] Front-end frameworks by popularity.
<https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190> (22.04.2021)
- [9] Krill P. Slim, speedy Svelte framework puts JavaScript on a diet, 2016.
<https://www.infoworld.com/article/3146966/slim-speedy-svelte-framework-puts-javascript-on-a-diet.html> (06.05.2021)
- [10] The State of CSS 2020: CSS-in-JS. <https://2020.stateofcss.com/en-US/technologies/css-in-js/> (27.04.2021)
- [11] NPM trends: Express vs Fastify. <https://www.npmtrends.com/express-vs-fastify> (27.04.2021)
- [12] npm, Inc. achieves record results in 2018, doubles down on enterprise strategy in 2019.
<https://apnews.com/press-release/pr-businesswire/ada6389643484a47932b7cf69cb450ce> (01.05.2021)
- [13] Flux: In-Depth Overview. <https://facebook.github.io/flux/docs/in-depth-overview> (06.05.2021)

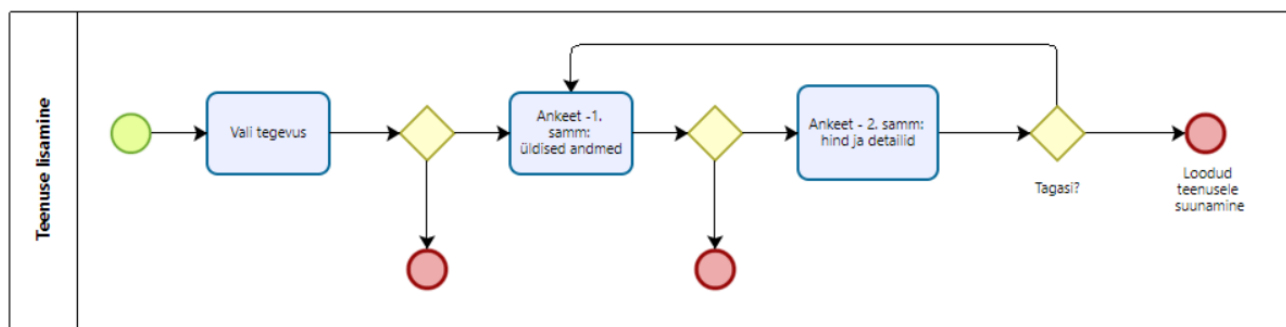
[14] Chen L. Micoservices: Architecting for Continuous Delivery and DevOps, 2018.
https://www.researchgate.net/publication/323944215_Microservices_Architecting_for_Continuous_Delivery_and_DevOps (06.05.2021)

Lisad

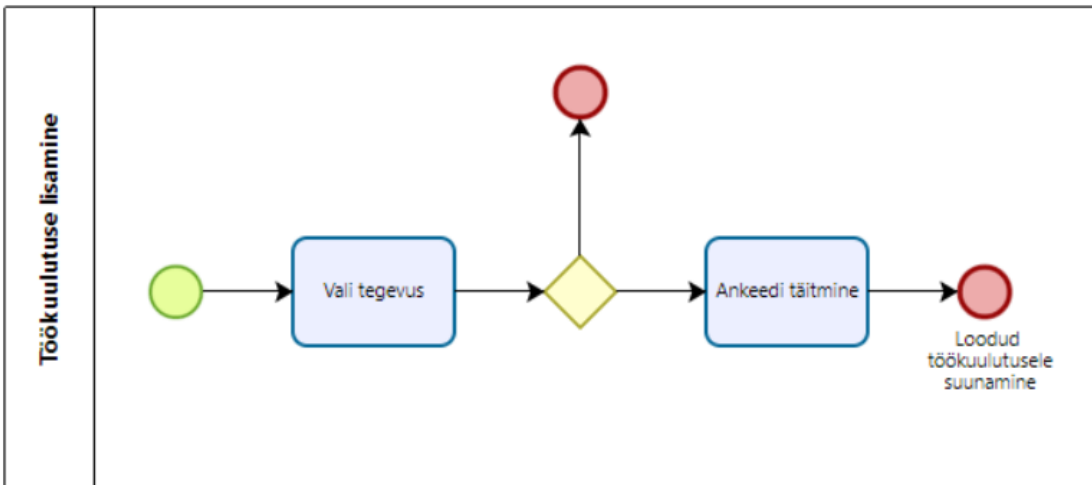
I. Autentimise voo diagramm



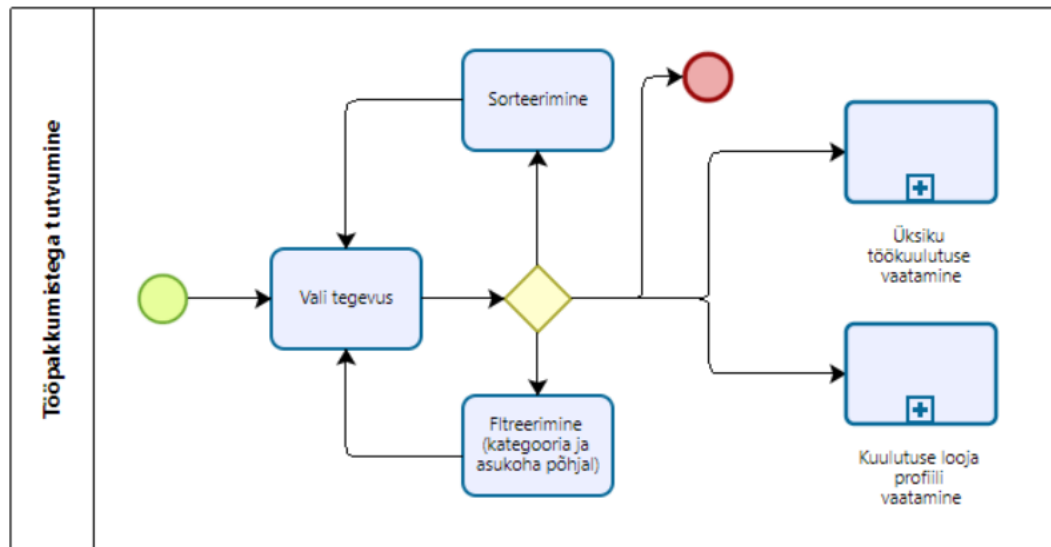
II. Teenuse reklaami lisamise voo diagramm



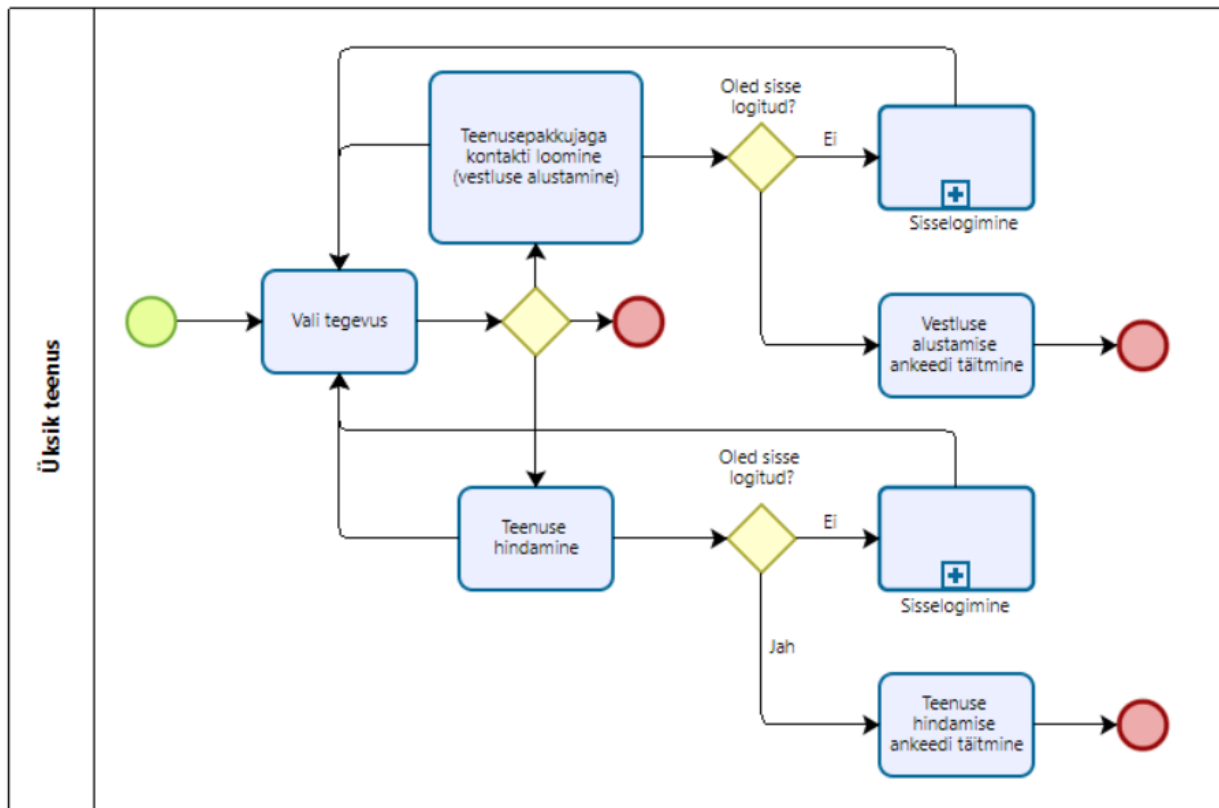
III. Töökuulutuse reklaami lisamise voo diagramm



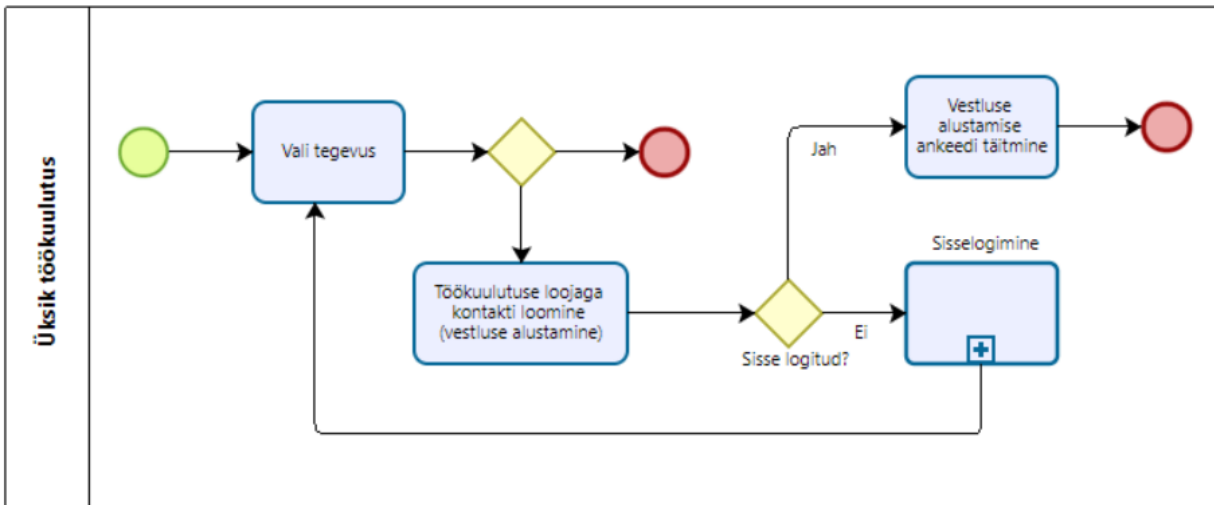
IV. Kataloogivaate diagramm



V. Üksiku teenuse voo diagramm



VI. Üksiku töökuulutuse voo diagramm



Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Rando Leppik**,

(autori nimi)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose
Veebirakenduse loomine ettevõttele Tööleidja,
(lõputöö pealkiri)

mille juhendaja on **Vambola Leping**,
(juhendaja nimi)

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 3.0, 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.

Rando Leppik
05.05.2021