

LATVIJAS UNIVERSITĀTE  
DATORIKAS FAKULTĀTE

**PERSONĪGĀS ZINĀŠANU SISTĒMAS**

BAKALAURA DARBS

Autors: **Svetlana Poļakova**

Studenta apliecības Nr.: sp12037

Darba vadītājs: asociētais profesors Dr. dat. Uldis Straujums

RĪGA 2016

## **Anotācija**

Darba tika izpētīts, kas ir personīgas zināšanas, personīgo zināšanu pārvaldība un personīgo zināšanu sistēmas, un kādas ir to svarīgākās funkcijas. Balstoties uz pētījumu, tika izstrādāts risinājuma prototips, izmantojot grafu datubāzi *neo4j* un *jQuery* bibliotēku. Tika definēts virziens tālākai sistēmas izstrādei.

**Atslēgvārdi:** personīgas zināšanas, personīgo zināšanu pārvaldība, grafu datubāze

## **Abstract**

### **Personal Knowledge systems**

The purpose of this work was to research what is personal knowledge, personal knowledge management and personal knowledge systems, and also define its main functions. Based on the research there was developed prototype of the system. It was created using *neo4j* graph database and *jQuery* language. Author defined direction of the future work.

**Keywords:** personal knowledge, personal knowledge management, graph database

## Saturs

<b>APZĪMĒJUMU SARAKSTS</b> .....	5
<b>IEVADS</b> .....	6
<b>1. PERSONĪGĀS ZINĀŠANAS</b> .....	7
<b>1.1. Kas ir personīgās zināšanas</b> .....	7
<b>1.2. Personīgo zināšanu pārvaldība</b> .....	7
<b>1.3. Personīgo zināšanu sistēmas</b> .....	9
<b>1.3.1. Zināšanu attēlošana</b> .....	9
<b>1.3.2. Semantiskais tīkls</b> .....	10
<b>1.3.3. Zināšanu grafs</b> .....	10
<b>1.3.4. RDF</b> .....	11
<b>2. PERSONĪGO ZINĀŠANU SISTĒMU APRAKSTS</b> .....	12
<b>2.1. Datu tipi</b> .....	12
<b>2.1.1. Brīvais teksts</b> .....	12
<b>2.1.2. Saite</b> .....	12
<b>2.1.3. Nosaukums, vārds vai īsa frāze</b> .....	12
<b>2.1.4. Attēls, video vai audio</b> .....	12
<b>2.1.5. Salikts objekts</b> .....	13
<b>2.2. Datubāzes tipi</b> .....	13
<b>2.2.1. Relāciju datubāzes</b> .....	13
<b>2.2.2. Grafu datubāzes</b> .....	14
<b>2.3. Personīgas wiki</b> .....	15
<b>3. EKSISTĒJOŠIE RISINĀJUMI</b> .....	16
<b>3.1. Evernote</b> .....	16
<b>3.2. OneNote</b> .....	16
<b>3.3. Google Disks</b> .....	17
<b>3.4. DEVONthink</b> .....	17
<b>4. PERSONĪGO ZINĀŠANU SISTĒMU UZBŪVE</b> .....	19
<b>4.1. Funkcionālas prasības</b> .....	19
<b>4.1.1. Iespēja pievienot brīva teksta ierakstu</b> .....	19
<b>4.1.2. Ērta meklēšana</b> .....	19
<b>4.1.3. Uzdevumu pārvaldība</b> .....	19
<b>4.1.4. Kalendārs</b> .....	20
<b>4.1.5. Iespēja vilkt un nomest datnes</b> .....	20

4.1.6.	Iespēja izveidot domu kartes.....	21
4.2.	Nefunkcionālas prasības.....	22
4.2.1.	Visi dati vienā vietā.....	22
4.2.2.	Vairāki datu tipi.....	22
4.2.3.	Vairāku platformu atbalsts.....	22
4.2.4.	Datu sinhronizācija un drošība.....	22
4.3.	Datu struktūra un organizācija.....	23
4.4.	Datu glabāšana.....	23
4.5.	Datu attēlošana.....	23
5.	AUTORA RISINĀJUMS.....	24
	REZULTĀTI UN DISKUSIJA.....	26
	SECINĀJUMI.....	27
	IZMANTOTĀ LITERATŪRA UN AVOTI.....	28
	Pielikums 1. Darbības ar neo4j datubāzi.....	30

## **APZĪMĒJUMU SARAKSTS**

jQuery – JavaScript valodas bibliotēka

Neo4j – grafu datubāze

RDF (Resource Description Framework) – standarta modelis datu attēlošanai tīmeklī

XML (Extensible Markup Language) – W3C rekomendācija speciālas nozīmes  
iezīmēšanas valodu veidošanai

Linux – Unix-bazēta operētājsistēma

## **IEVADS**

Dzīvojot informācijas laikmetā, cilvēkiem ikdienā jāstopas ar lieliem informācijas apjomiem. Tā kā cilvēku atmiņa ir ierobežota, pastāv nepieciešamība saglabāt un pārvaldīt datus, ar ko arī nodarbojas zināšanu pārvaldība. Personīgās zināšanās, t.i. cilvēka atmiņa, pieredze iespaidi un idejas arī pieprasa pārvaldību. Personīgo zināšanu sistēmu jautājums tiek pētīts vairāk nekā 70 gadus, un līdz šim tā arī netika atrasts perfekts risinājums.

Šī darba mērķis ir noskaidrot, kas ir personīgas zināšanas un personīgo zināšanu sistēmas, apskatīt eksistējošus risinājumus šīm sistēmām, novērtēt to galvenās īpašības un funkcijas, un izveidot savas sistēmas prototipu.

# 1. PERSONĪGĀS ZINĀŠANAS

## 1.1. Kas ir personīgās zināšanas

Personīgās zināšanas ir visas zināšanas, kas tika apgūtas personīgi, balstoties uz savu pieredzi, personīgo iesaisti, nevis uz lasītu vai dzirdētu informāciju [1].

Piemēram, kad mēs lasām par vēsturiskiem notikumiem, kurus paši nepiedzīvojam, tad mēs redzam tikai plikus faktus vai citu cilvēku interpretāciju. Šajā gadījumā mūsu viedoklis tiks balstīts uz svešu cilvēku viedokli vai stāstu, nevis uz savu pieredzi. Priekš mums šīs zināšanas nav personīgas. Savukārt, visas gleznas, kuras mēs redzējam ar savām acīm, visa mūzika, kuru mēs dzirdējam, visas atmiņas, iespaidi un idejas pieder pie personīgām zināšanām.

Personīgas zināšanas ietekmē cilvēka unikālā pieredze, gaume, izpratne un interese par jautājumu, tāpēc to bieži ir grūti paskaidrot citiem. Pirmkārt, to pašu informāciju viņi uztvers savādāk, balstoties uz iepriekš dzirdēto un redzēto. Otrkārt, ne visas lietas var paskaidrot ar vārdiem. Var simts reizes pastāstīt, ka sarkana krāsa ir spilgta, un silta, un asociējas ar kaut ko bīstamu, bet kamēr cilvēks sarkanu krāsu neredzēs, visticamāk, iedomāties to viņam nesanāks.

Personīgās zināšanas ir subjektīvas, un var nebūt svarīgas vai pat patiesas citai personai. Piemēram, manas subjektīvas zināšanas ir ka zaļās olīvas ir garšīgas, bet melnas – nav, bet var atrast diezgan daudz cilvēkus, kam garšo melnas olīvas, vai negaršo zaļas.

Personīgās zināšanas var nosaukt par vissvarīgākajām cilvēka zināšanām.

## 1.2. Personīgo zināšanu pārvaldība

Zināšanu pārvaldība ir procesu kopa, kas iekļauj sevī zināšanu formulēšanu, organizāciju, attēlošanu, plānošanu, kā arī to pielietošanu [11]. Visbiežāk, runājot par zināšanu pārvaldību, runa ir par organizācijas zināšanām.

Zināšanu pārvaldība ir cieši saistīta ar kognitīvo zinātņu nozari, psiholoģiju un mākslīgo intelektu. Problēmas, kuras risina zināšanu pārvaldība ir saistītas

IT jomu nevar iedomāties bez zināšanu pārvaldības rīkiem. Piemēram, JIRA un Mantis piedāvā plašu funkcionalitāti darbinieku vajadzībām, lai nodrošinātu galveno mērķi – uzdevumu pārvaldību un izpildi.

Pretēji klasiskai zināšanu pārvaldībai, kur uzsvars ir uz sadarbību ar citiem, personīgo zināšanu pārvaldības mērķis ir organizēt vienas personas darbu, laiku vai personālo attīstību, un ļaut viņam vai viņai sasniegt savu mērķus. Daudzu personīgo zināšanu sistēmu nolūks ir cilvēka apmācība [12].

Personīgo zināšanu pārvaldību var uzskatīt kā prasmju kopu, kuras nolūks ir risināt problēmas, pieņemt lēmumus utt. Frands un Hiksons (1999) uzskatīja par svarīgiem sekojošās prasmes:

- Informācijas meklēšana.
- Informācijas novērtēšana – atrast tikai svarīgu lietotājam informāciju, filtrēt informāciju pēc noteiktiem kritērijiem.
- Informācijas organizēšana – atgriezt informāciju pārskatāmā formātā, piemēram, pēc datuma vai pēc kādas citas īpašības.
- Informācijas analīze – vai no dotās informācijas var izteikt atzinumu, iegūt jauno informāciju.
- Informācijas attēlošana.
- Informācijas aizsargāšana – nodrošināt, lai informācija netiktu nodota citu personu rokās.
- Sadarbība ap informāciju – nodrošināt iespēju dalīties ar informāciju, strādāt kopā ar citu cilvēku vai cilvēku grupu [2].

Mūsdienās personīgo zināšanu pārvaldības nozīme ir nedaudz mainījies – šobrīd tā ir mazāk saistīta ar darbu, un vairāk – ar personīgām interesēm un attīstību, un iekļauj sevī ideju ražošanu un zināšanu pielietošanu dzīvē.

Zināšanu pārvaldība ir darbietilpīgs process, tāpēc ir vērts izstrādāt sistēmas, lai to optimizētu.

### 1.3. Personīgo zināšanu sistēmas

1945. gadā Vanevers Bušs savā esejā “Kā mēs varam domāt” piedāvāja aprakstu sistēmai “Memex”, kurā cilvēks varētu saglabāt savas grāmatas, kontaktus un pierakstus, t.i. to, ko mēs varam nosaukt par personīgām zināšanām. “Memex” bija pirmais personīgo zināšanu sistēmas prototips. “Memex” tā arī netika implementēts tādā veidā, kā bija aprakstīts esejā, tomēr tas ietekmēja pasaules tīmekļa attīstību, kā arī pirmo personīgo zināšanu sistēmu izveidi [5].

Neviena no līdz šim izstrādātajām personīgo zināšanu sistēmām nav universāla, un neviena nesatur sevī visas nepieciešamās funkcijas. Daudzas eksistējošās programmas ir novecojušas, un vairs neatbilst mūsdienu prasībām.

#### 1.3.1. Zināšanu attēlošana

Svarīgs jautājums personīgo zināšanu sistēmu kontekstā ir zināšanu attēlošana. Mākslīgā intelekta nozarē zināšanu attēlošana ir metožu kopa, kas ļauj iekodēt un ievietot zināšanas kādas sistēmas zināšanu bāzē. Zināšanu attēlošanas pamatuzdevums ir iemācīties organizēt un strukturēt resursus un datus tā, lai datorprogrammas varētu apstrādāt tos bez cilvēku iesaistes, līdzīgi cilvēku intelektam.

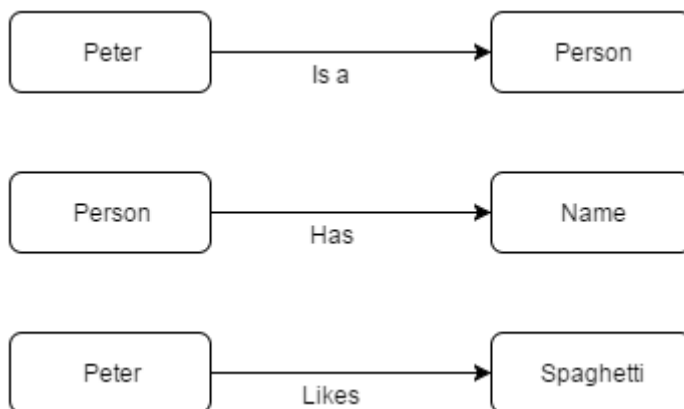
Pareizi izvēlētā zināšanu attēlošana var atvieglot daudzu sarežģīto problēmu risinājumu.

Ir četri zināšanu attēlošanas shēmu veidi:

- Loģiskās shēmas attēlo zināšanas ar matemātisko vai ortogrāfisko simbolu palīdzību, kā arī secināšanas likumiem un precīzi noteikto sintaksi un semantiku.
- Procesuālas shēmas – zināšanas ir attēlotas kā instrukciju kopa problēmu risināšanai.
- Tīkla shēmas attēlo zināšanas kā grafu, kur virsotnes ir zināšanu objekti un šķautnes nosaka attiecības un saites starp tām.
- Strukturētas shēmas ir tīkla shēmu paplašinājums, kur katra virsotne ir saliktais objekts ar dažādām īpašībām [6].

### 1.3.2. Semantiskais tīkls

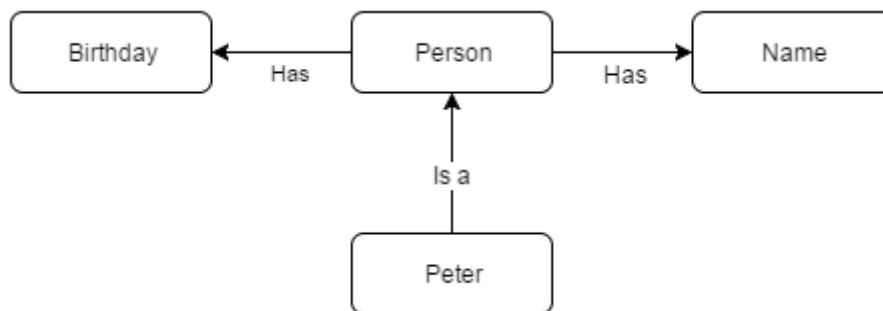
Viens no zināšanu attēlošanas veidiem ir semantiskais tīkls. Semantiskais tīkls ir grafa paveids, kur mezgli ir objekti vai koncepti, kas tiek savienoti ar attiecībām [6].



att. 1.1. Attiecības starp objektiem

Attēlā 1.1. ir parādīts, ka attiecības starp objektiem var norādīt gan uz objekta atribūtiem, gan uz atribūtu vērtībām, gan uz to, ka viens objekts ir instance no otra.

Semantiskais tīkls atbalsta mantošanas principu un prototipus, kad objekts manto savas īpašības no “vecāka”.



att. 1.2. Īpašību mantošana objektiem

Attēlā 1.2. ir redzams, ka instance “Peter” manto īpašības no vecāka “Person”.

### 1.3.3. Zināšanu grafs

Zināšanu grafs kā zināšanu attēlošanas veids tika attīstīts no tā saucamā konceptuālā grafa. Tāpat kā semantiskajā tīklā, koncepti ir attēloti kā mezgli, un savienoti ar saitēm, kas nosaka attiecības starp konceptiem [7].

Zināšanu grafu teorija nodarbojās ar valodu pārsēšanu datoriem saprotamā veidā un to apstrādi. Iegūtas zināšanas tiek izmantotas dokumentu apstrādē un kategorizācijā.

#### 1.3.4. RDF

RDF (Resource Description Framework) ir abstrakts modelis datu attēlošanai tīmeklī, kas nosaka datu struktūru, kā arī veidu, kā apstrādāt datus [4].

RDF kontekstā varīgākie jēdzieni ir resursi, īpašības un apgalvojumi. Par resursiem var būt uzskatīti jebkuri objekti vai jēdzieni. Katram resursam tiek piešķirts URI (Unikāls Resursu Identifikators), par kuru bieži tiek izmantota resursa adrese tīmeklī (URL).

Katram resursam ir dažādas īpašības, kas ir atkarīgas no resursu tipa. Piemēram, personai pamatīpašības būtu “vārds”, “dzimšanas datums”, “vecāki”, “bērni”, “intereses” utt.

Apgalvojumi ir domāti resursu savienošanai. Apgalvojums sastāv no trim daļām, kas ir objekts, predikāts un subjekts. Vairāki apgalvojumi par vienu un to pašu resursu var veidot semantisko tīklu.

RDF nolūks ir strukturēt datus tā, lai tie varētu būt izmantoti lietotnēs. Pats pirmais RDF serializācijas standarts tika bāzēts uz XML un tika nosaukts par RDF/XML. Eksistē arī JSON-LD (JSON-bāzēts), RDFa (izmantojams HTML datnēs) un Turtle ģimenes formāti. Kaut arī tie izmanto dažādus pierakstu veidus, funkcionāli tie neatšķiras [8].

RDF pamatmērķis ir strukturēt datus tā, lai būtu iespējams automātiski savienot vairākus informācijas avotus vienā, un lai iegūta informācija būtu lietderīga. Lai to izdarītu, tika izdomāts vienots datu struktūras stils, kā arī apgalvojumu stils.

## **2. PERSONĪGO ZINĀŠANU SISTĒMU APRAKSTS**

Šī nodaļā tiek aprakstīta personīgo zināšanu sistēmu uzbūve.

### **2.1. Datu tipi**

Tika apskatīti populārākie datu tipi, kas tiek vai var būt izmantoti personīgo zināšanu sistēmās.

#### **2.1.1. Brīvais teksts**

Pats vecākais un pats vienkāršākais datu tips ir brīvais teksts vai teksta dokuments. Tas tiek izmantots gandrīz visās personīgo zināšanu sistēmās. Brīvais teksts ir izmantots gan kā objektu apraksts, gan kā patstāvīgs ieraksts datubāzē, piemēram, ja cilvēks raksta blogu. Brīvais teksts ir vecākais no datu tipiem, un eksistē gandrīz katrā no personīgo zināšanu sistēmām.

#### **2.1.2. Saite**

Saite kā datu tips ir nepieciešama, lai veidotu norādes uz resursiem, kas ir nepieciešamās personīgo zināšanu sistēmā, bet kurus nav iespējams vai nav jēgas kopēt teksta vai attēlu veidā.

#### **2.1.3. Nosaukums, vārds vai īsa frāze**

Viena vārda vai frāzes objekts bieži tiek izmantots, veidojot domu kartes vai koncepta kartes. Šīm objektam ir ierobežots simbolu skaits, tam ir jābūt īsam, saturīgam un viennozīmīgam. To ir vērts izmantot, veidojot skices un diagrammas, savienojot dažādus objektus ar vienu ideju vai konceptu.

#### **2.1.4. Attēls, video vai audio**

Tā kā attēli, shēmas, video un audioieraksti pieder pie personīgām zināšanām, sistēmas atļauj glabāt un sakārtot arī tos. Bieži tiek glabāta ne pati datne, bet tikai saite uz to.

### **2.1.5. Salikts objekts**

Bieži vien, taisot sarežģītu sistēmu, ar vienu tekstu neietiek. Ir daudz ērtāk apstrādāt salikto objektu ar vairākiem atribūtiem. Piemēram, persona ir salikts objekts, kuram ir atribūti ir vārds, dzimšanas datums, personas kods, dzīvesvieta, e-pasts, telefona numurs, dažādas intereses utt.

## **2.2. Datubāzes tipi**

Viens no veidiem, kā kategorizēt personīgas zināšanu sistēmas, ir sadalīt tās pēc datubāzes tiem.

### **2.2.1. Relāciju datubāzes**

Klasiskās relāciju datubāzes tiek izmantotās sistēmas, kas izmanto datu kategorizāciju. Šajās sistēmās uzsvars ir uz datu grupēšanu pēc kāda principa, nevis uz attiecību noteikšanu. Relāciju datubāzes ir labi izpētītas, daudzas funkcijas ir iekļautas pēc noklusējuma, un ir iespējams implementēt gandrīz jebkuru ideju un rēķināties uz risinājuma drošību.

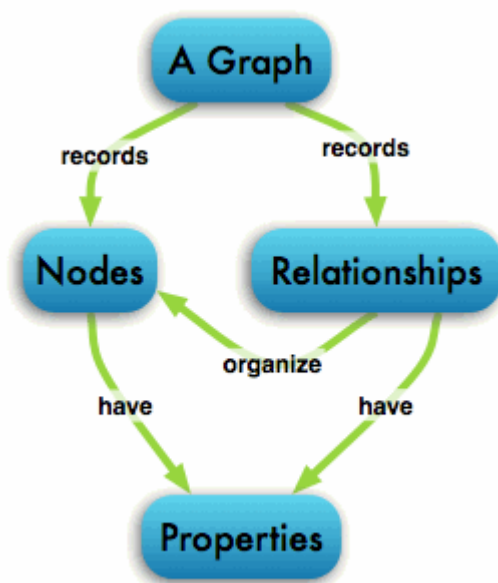
Relāciju datubāzes tiek izmantotas gadījumos, kad sistēmā ir ierobežots datu tipu skaits. Piemēram, Evernote objekti ir “pieraksti” vai “pierakstu grāmatas” – tos ir viegli pārvaldīt, lietojot klasisko datubāžu veidu. Arī blogu ieraksti visbiežāk izmanto relāciju datubāzes.

Saites starp ierakstiem relāciju datubāzēs parasti tiek risinātas ar ārējo atslēgu palīdzību, vai arī ar starptabulām. Ārējas atslēgas norāda uz cita ieraksta primāro atslēgu – unikālo identifikatoru. Savukārt starptabulas ir atsevišķas tabulas, kur tiek ierakstīti divi vai vairāki unikālie identifikatori, kas norāda uz objektiem, kā arī ir iespējams norādīt papildus informāciju, piemēram, attiecību veidu.

### 2.2.2. Grafu datubāzes

Ja sistēmas pamatuzdevums ir noteikt attiecības starp dažādiem objektiem, vai zīmēt domu kartes vai jēdzienu kartes, daudz ērtāk ir izmantot grafu datubāzes. Dažiem pieprasījumiem, tāds grafu datubāžu rīks kā Neo4j ir parādījis labākus ātrdarbības rezultātus, nekā relāciju datubāzes [13].

Grafa mezgli ir ierakstu vienības, kas tiek savienoti ar novirzītām saitēm, kas nosaka attiecības starp ierakstiem. Attiecības grafu datubāzēs ir līdzīgas ierakstiem starptabulās. Saites nevar eksistēt paši par sevi un norādīt uz tukšumu, tiem obligāti jābūt “starp” diviem mezgliem. Dzēšot mezglu, vienmēr dzēšas arī visas attiecīgas saites [10]. *Attēlā 2.1.* ir redzama grafa struktūra.



*att. 2.1. Grafu datubāzes struktūra*

Grafu datubāzes ir viegli savienojamas ar RDF datiem.

Galvenais grafu datubāžu trūkums ir vājāka drošība – bieži tā nav iebūvētā, atšķirībā no SQL datubāzēm, un par to izstrādātājam bieži ir jārūpējas pašam.

### 2.3. Personīgas wiki

Wiki ir rīki vai tīmekļa vietnes, kas ļauj lietotājiem pievienot hiperteksta lapas un rediģēt un pilnveidot lapā esošo informāciju [15]. Viena no wiki priekšrocībām ir hipersaites starp vairākiem dokumentiem vienā vietnē.

Pēc koncepta wiki jābūt pēc iespējas vienkāršākiem, un neprasīt no lietotāja nekādas īpašas prasmes vai programmatūras.

Personīgās wiki, kā seko no nosaukuma, ir izveidoti personīgai lietošanai. Daži no risinājumiem atbalsta wiki satura publicēšanu tīmeklī, kur iespēja rediģēt ir iedota tikai vienam autoram, bet citi lietotāji var lasīt publicēto. Daži risinājumi glabā datus lokāli vai aizsargā datus ar paroli, noliedzot piekļuvi visiem, izņemot autoru.

Personīgas wiki, līdzīgi tradicionāliem, atbalsta meklēšanu pēc lapas nosaukuma. Daži no risinājumiem atbalsta meklēšanu arī pēc lapas satura.

### **3. EKSISTĒJOŠIE RISINĀJUMI**

Šī nodaļā tiek apskatītas un salīdzinātas populārākās un nozīmīgākās personīgo zināšanu sistēmas. Tās tika izvēlētas, balstoties gan uz autora personīgo pieredzi, gan uz lietotņu popularitāti tīmeklī.

#### **3.1. Evernote**

Evernote ir multi-platformu rīks, kas ļauj saglabāt piezīmes, augšupielādēt attēlus vai audiodatnes [17]. Katrai teksta piezīmei ir nosaukums un formatēta teksta daļa. Jebkurai piezīmei var pielikt teksta birku un atgādinājumu, izvēloties konkrēto datumu. Piezīmes var sadalīt pa kategorijām – piezīmju grāmatām. Evernote piedāvā arī iespēju veidot uzdevumu sarakstu, kur katru no uzdevumiem var iezīmēt kā “izpildītu”.

Papildus iespēja ir “saglabāt daļu no tīmekļa lapas”. Ar speciālo rīku iezīmējot daļu no vietnes, var saglabāt to, lai pēc tam nenāktos meklēt, vai lai būtu piekļuve tai arī bezsaistē.

Rīks ir pieejams gan kā tīmekļa vietne, gan kā viedtālruna lietotne, gan kā datorprogramma. Eksistē versijas populārākām ierīcēm un operētājsistēmām.

Salīdzinoši ar citām apskatītām sistēmām var teikt, ka Evernote ir par maz funkcijas, kaut arī izpildījums ir ērts un ātrs.

#### **3.2. OneNote**

OneNote ir Microsoft produkts, kas ir pieejams gandrīz jebkurā populārākā platformā, izņemot Linux OS. Eksistē arī tīmekļa versija [18].

OneNote ļauj lietotājiem veikt pierakstus, veidot uzdevumu sarakstus. Ir ļoti ērti veidot domu kartes, tabulas, zīmēt un dzēst detaļas. Ir diezgan ērta datu sadalīšana pa mapēm, cilnēm un lapām.

Nav paredzēta iespēja veidot personīgo wiki vai blogu, bet ir datu koplietošanas iespēja.

Ņemot vērā, ka OneNote iet kopā ar citiem Microsoft Office produktiem, kopā sanāk samēri ērta sistēma.

### **3.3. Google Disks**

Google Disks ir bezmaksas rīks datņu izveidei un glabāšanai [16]. Programmai eksistē gan tīmekļa versija, gan datora un viedtālruņa lietotnes. Datnes glabājās dažādos formātos, ir iespējams glabāt teksta dokumentus, attēlus, izveidot mapes, kā arī piesaistīt papildus lietotnes dažādu shēmu veidošanai un datņu apstrādei.

Viena no svarīgākajām funkcijām Google Diska dokumentiem ir ērta un droša koplietošana – var noteikt, kurš var datni apskatīties, un kurš drīkst to rediģēt.

Viena no problēma ir tas, ka nav iespējams sakārtot mapes un datnes tā, kā lietotājam būtu ērtāk, kaut arī Google Disks atbalsta drag & drop funkcionalitāti – ir iespējams sakārtot tikai pēc datuma un pēc nosaukuma.

Arī redaktoram trūkst daudzas nozīmīgas īpašības, kas ir implementētas vispopulārākajos teksta redaktoros (Microsoft Word, OpenOffice Writer, LibreOffice Writer), tāpēc nevar tos aizvietot. Modificējamām datnēm ļoti bieži atšķiras formatējums, atvērot tās ar citu programmu Dažas papildus lietotnes, kas var tikt izmantotas shēmu un diagrammu veidošanai strādā samēri lēni, un nav ērti lietojamas.

Var secināt, ka Google Disks nav perfekts, bet tomēr ir apmierinošs risinājums, pat neskatoties uz to trūkumiem.

### **3.4. DEVONthink**

DEVONthink lietotne pozicionē sevi kā zināšanu pārvaldības rīks [19]. Tam eksistē vairākas versijas ar dažādu maksu. Bezmaksas ir tikai izmēģinājuma versija.

Lietotnei ir implementēts diezgan ērts veids kā glabāt un organizēt datnes, kā arī eksistē iespēja uztaisīt savu personīgo wiki.

Galvenais šīs sistēmas trūkums ir tas, ka tas strādā tikai uz Mac OS. Tīmekļa versijas nav. Dažādas funkcijas, kā, piemēram, e-pasta sinhronizācija, ir pieejamas tikai par papildus maksu.

Neskatoties uz to, ka vairāki lietotāji tīmeklī to uzskata par samēri ērtu sistēmu, tai trūkst daudzas priekšrocības, kas ir citām sistēmām.

## **4. PERSONĪGO ZINĀŠANU SISTĒMU UZBŪVE**

Šī nodaļā tiek apskatīts, kā labāk būtu veidot personīgi zināšanu sistēmas, tās pamatfunkcijas un galvenās vajadzības. Protams, nekādas viennozīmīgas pamācības tam neeksistē, jo katram cilvēkam ir savs uzskats par to, kādai ir jābūt zināšanu pārvaldības sistēmai, bet es mēģināju noteikt galvenās kopējas vadlīnijas.

### **4.1. Funkcionālas prasības**

Funkcionālas prasības nosaka, kādas funkcijas piemīt personīgo zināšanu sistēmām.

#### **4.1.1. Iespēja pievienot brīva teksta ierakstu**

Iespēja pievienot brīva teksta ierakstu ir noteikti svarīgākā no personīgo zināšanu sistēmas funkcijām. Iespēja neierobežoti formulēt domas, formatēt tās un iezīmēt galveno, nozīmē iespēju saglabāt savas atmiņas par piedzīvojumiem, pārdzīvojumiem, un iespaidiem, kā arī dažreiz dalīties ar citiem.

#### **4.1.2. Ērta meklēšana**

Ērta informācija vai datnes meklēšana ir viena no svarīgākajām sistēmas funkcijām. Meklēšanai jāvar norādīt vairāki parametri, piemēram, meklēt nosaukumā, vai tekstā, vai abos, rezultātiem jābūt sakārtotiem pēc noteiktiem kritērijiem – pēc izveidošanas/redīgēšanas datuma, pēc popularitātes.

#### **4.1.3. Uzdevumu pārvaldība**

Lietotājam jābūt iespējai veidot sarakstus ar uzdevumiem, kā arī norādīt uzdevumu statusu: “izpildīts”, “neizpildīts”, “izpildīšanas procesā” utt. Ir labi, ja

uzdevumam var pievienot arī gala termiņu. Ir arī daudz ērtāk, ja uzdevumus var sadalīt pa kategorijām.

#### **4.1.4. Kalendārs**

Tā kā viena personīgo zināšanu sistēmas mērķiem ir efektīva darba un personīga laika plānošana, sistēmai jāsaturs kalendārs, kurā varētu ievietot svarīgus datumus, notikumus, un termiņus. Vislabāk, ja kalendāram būtu atvēlēts atsevišķais bloks vai pat skats, kurā varētu ne tikai apskatīties esošos notikumus, bet arī rediģēt tos un pievienot jaunus.

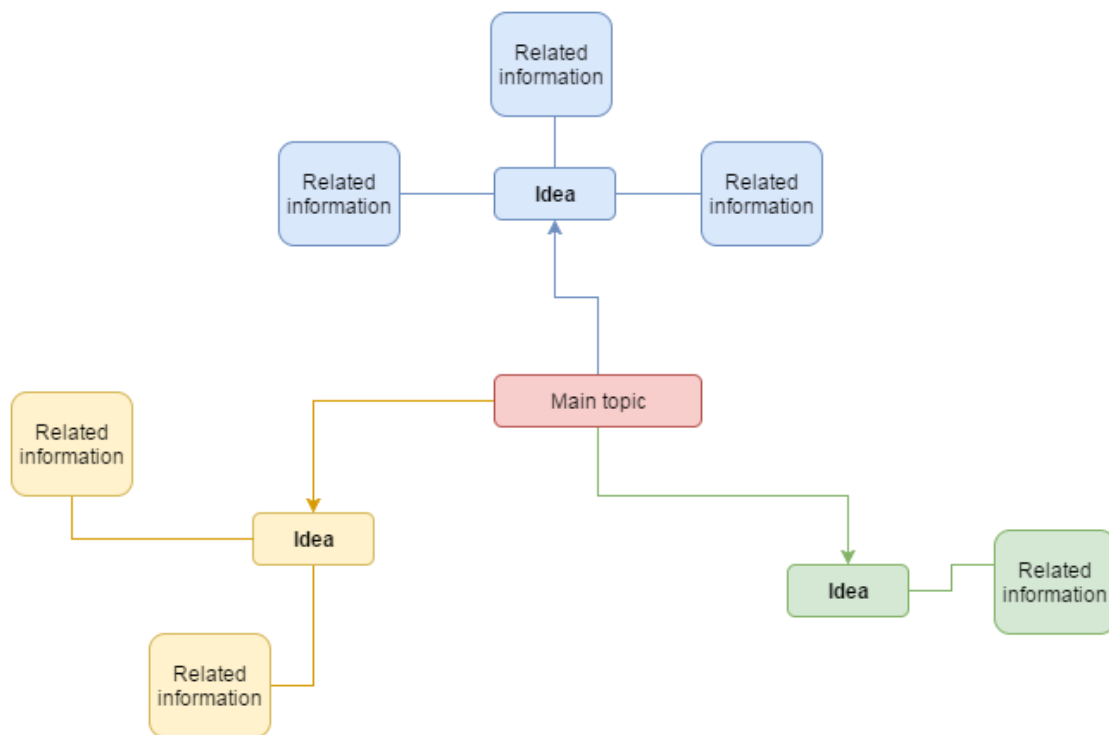
#### **4.1.5. Iespēja vilkt un nomest datnes**

Vilkt un nomest datnes (drag & drop) funkcija ļoti atvieglo datnes pievienošanu sistēmai.

#### 4.1.6. Iespēja izveidot domu kartes

Domu kartes ir diagrammas, kas attēlo attiecības starp procesiem vai uzdevumiem, kas ir apvienoti ar kādu kopējo mērķi vai ideju.

Kā tika pierādīts kognitīvās zinātnēs, domu kartes palīdz atrast sakarības un cēloņus procesiem, apskatīt situāciju kopumā, atrast trūkstošus elementus, kā arī atcerēties informāciju.



att. 4.1. Domu karte

Domu un konceptu kartes ir svarīga daļa no personīgo zināšanu pārvaldības, tāpēc sistēmai jābūt iespējai izveidot tās.

## **4.2. Nefunkcionālas prasības**

Nefunkcionālas prasības apraksta sistēmu no lietojamības skata, nosaka, kādai sistēmai ir jābūt, nevis ko tā darīs. Pati galvenā īpašība, kas piemīt labai personīgo zināšanu sistēmai, ir tas, ka lietotāji šo sistēmu uzskata par ērtu.

### **4.2.1. Visi dati vienā vietā**

Viena no svarīgām personīgo zināšanu sistēmu īpašībām ir tas, lai visi dati, ko lietotājs gribētu saglabāt, glabātos vienā sistēmā. Reti kuram būtu ērti vienā lietotnē vai tīmekļa vietnē saglabāt dokumentus, citā – attēlus, vēl citā – veidot domu kartes vai aizpildīt kalendāru, katru dienu tērēt laiku uz to, lai pārslēgtos starp resursiem un lietotnēm. Lietotāji labprāt izvēlētos vienu sistēmu ar plašāku funkcionalitāti.

### **4.2.2. Vairāki datu tipi**

Sistēmai būtu jāatbalsta ne tikai teksta, bet arī attēlu un audiodatņu ievade. Vēlams, lai katram datu tipam varētu apskatīties un rediģēt īpašības, kā arī jābūt iespējai norādīt attiecības starp vairākām datnēm.

### **4.2.3. Vairāku platformu atbalsts**

Sistēmai vismaz būtu jābūt tīmekļa vietnei, kas atbalstītu gan datora, gan viedtālrunu un planšetes versijas. Gadījumā, ja personīgo zināšanu sistēmai nav paredzēts savienojums ar internetu, labāk būtu izstrādāt datora un mobilās lietotnes. Sistēmas funkcionalitātei un izskatam jābūt neatkarīgam no ierīces operētājsistēmas.

### **4.2.4. Datu sinhronizācija un drošība**

Gadījumā, ja lietotne ir paredzēta vairākām ierīcēm, starp tām jānotiek datu sinhronizācijai. Tajā pašā laikā jānodrošina, lai nevienai svešai personai nebūtu piekļuve personas datiem.

Būtu jābūt arī iespējai uztaisīt sistēmas rezerves kopiju.

### **4.3. Datu struktūra un organizācija**

Datu struktūra un organizācija ir atkarīga no sistēmas nolūka. Ja sistēmas uzsvars ir uz līdzīgu datu glabāšanu, kā, piemēram, personīgais wiki vai bloga ieraksti, tad vieglāk un prātīgāk būtu izmantot labi izpētīto un viegli implementējamo relāciju datubāzi.

Gadījumā, ja sistēmas uzsvars ir uz dažāda formāta datu savienošanu vienā tīklā, prātīgi būtu implementēt grafu datubāzi.

### **4.4. Datu glabāšana**

Visbiežāk izmantots veids, kā glabāt datus ir glabāt tos uz servera. Servera pusē tiek nodrošināta datubāze, un lietotājs pats izvēlās, ko glabāt sistēmā. Visbiežāk tās ir sistēmas, kam eksistē tikai tīmekļa versija. Liels trūkums ir tas, ka šādas sistēmas nevar strādāt bez interneta savienojuma.

Dažas sistēmas nodrošina ne tikai datņu augšupielādi uz servera, bet arī tos sinhronizāciju ar noteiktu mapi uz datora vai uz viedtālruni. Šāds veids nodrošina datu rezerves kopēšanu, un ļauj ērti pārvietot datnes no vienas ierīces uz otru. Lielāka daļa no šī veida sistēmām atļauj darbības ar datnēm arī bez interneta savienojuma.

Cits veids ir glabāt datus lokāli. Bez pieslēgšanas pie interneta sistēma daudz paliek drošāka, bet pastāv iespēja zaudēt visus datus reizē.

### **4.5. Datu attēlošana**

Datu attēlošana sistēmā ir ļoti atkarīga no datu tipa. Jo lielākā un sarežģītākā ir sistēma, jo vairāk uzmanības jāpievērš datu attēlošanas labākai praksei.

Ja sistēma atbalsta datņu glabāšanu, visērtāk datnes un mapes ir apskatīties jau sen sevi pieradītā koka struktūrā.

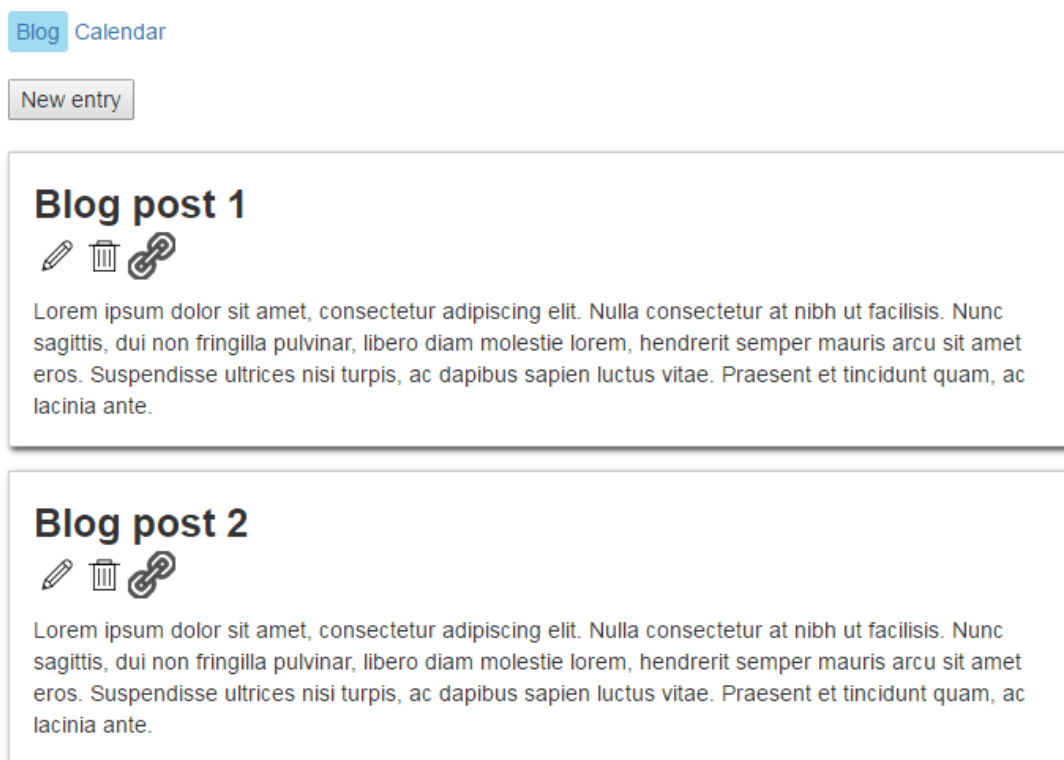
Kaut arī datu attēlošanas veidu sistēmas autori var izvēlēties paši, un par dizaina ērtību grūti atrast vienu pareizo viedokli, ir stingri ieteicams saglabāt iespēju strukturēt datus, ja tas ir iespējams, piemēram, apskatīties ierakstus tabulu vai grafu formātā.

## 5. AUTORA RISINĀJUMS

Darba izstrādes gaitā tikai izveidots risinājuma prototips, kas satur sevī nelielu daļu no iespējamām funkcijām. Izveidotās sistēmas primārais nolūks bija izveidot grafu ar savienotiem datiem. Tātad, galvenās funkcijas ir objektu pievienošana un savienošana savā starpā.

Sistēma tika izstrādāta kā tīmekļa vietne, izmantojot *jQuery* bibliotēku funkcionalitātes izveidei un *neo4j* datubāzi. *Neo4j* ir populārākā grafu datubāze, kas izmanto *Cypher* pieprasījumu valodu. Darbības ar datubāzi notiek ar *ajax* pieprasījumiem [20, 21, 22].

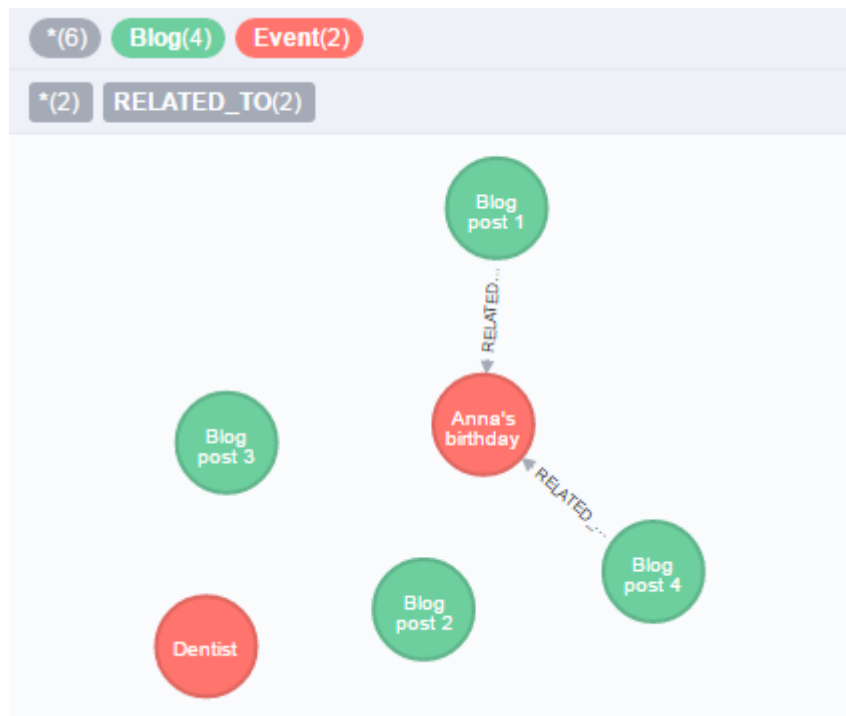
Šobrīd sistēma ļauj izveidot, dzēst un rediģēt bloga ierakstus un kalendārus notikumus. Bloga ierakstus var saistīt ar kalendāriem notikumiem.



att. 5.1. Izstrādātā prototipa ekrānuzņēmums

Izstrādātās sistēmas bloga skats ir redzams attēlā 5.1.

Attēlā 5.2. ir redzams sistēmas datubāzes grafa skats, kurā ir attēloti visi sistēmā esošie objekti un attiecības starp tiem. Sistēmai augot, palielināsies attiecību skaits



att. 5.1. Sistēmas datubāzes ekrānuzņēmums

Attēlā 5.3. mainīgais *query* ir uzrakstīts *Cypher* valodā. Vairāk kodu fragmentu var apskatīt pielikumā 1.

```
var query = {"query": "CREATE (:Blog { date: '"+new Date()  
+ "', title: '"+$blogTitle.val()  
+ "', text: '"+$blogTxt.val()  
+ "})", "params" : {}  
};
```

att. 5.3.. *Cypher* pieprasījuma piemērs

Izveidotā sistēma ir prototips, kā ir iespējams pārvaldīt zināšanas uz savienot tos savā starpā. Tālāk būtu ieteicams ļaut lietotājam pievienot vairākus datu tipus, iespējams, savienojoties ar Schema.org, vai arī ļaut veidot savus datu tipus.

Noteikti būtu jāļauj lietotājam apskatīties grafu tīmeklī, to var izveidot, lietojot *d3.js* bibliotēku.

## **REZULTĀTI UN DISKUSIJA**

Bakalaura darba izstrādes laikā tika izpētīts, kas ir personīgas zināšanas un personīgo zināšanu pārvaldība, to mērķi un problēmas.

Tika definētas dažas personīgo zināšanu sistēmu funkcijas un galvenās funkcionālās un nefunkcionālās vajadzības, tika apskatīti un novērtēti populārākie no jau eksistējošiem risinājumiem.

Darba ietvaros tika izstrādāts sava risinājuma prototips, kas piedāvā glabāt datus grafu datubāzē, savienojot tos savā starpā. Kaut arī autora risinājums nav perfekts, ir zināms virziens, kurā vajadzētu attīstīties. Darba izvirzītie mērķi tika sasniegti.

## SECINĀJUMI

Kaut arī personīgo zināšanu pārvaldības jautājums tiek pētīts jau vairāk par 70 gadiem, neviena no līdz šim izstrādātām sistēmām neatbilst pilnīgi visām personīgo zināšanu pārvaldības prasībām. Tika noskaidrots, ka prasības personīgo zināšanu sistēmām būtiski neatšķiras no prasībām priekš citām zināšanu sistēmām, kaut gan uzsvars ir uz personīgiem labumiem, nevis uz korporatīviem vai zinātniskiem.

Pēdējā laikā tika izveidotas jaunas pieejas personīgo zināšanu pārvaldības jautājumam, izstrādātāji atkāpjas no personīgo wiki koncepta, un parādās jauni risinājumi. Ir noteikti arī nākamie soļi savas sistēmas izstrādei.

## IZMANTOTĀ LITERATŪRA UN AVOTI

1. Polanyi, Michael. Personal Knowledge: Towards a Post-Critical Philosophy. University of Chicago Press. 1958. 37. lpp.
2. Raj Agnihotri, Marvin D. Troutt, (2009) "The effective use of technology in personal knowledge management: A framework of skills, tools and user context", Online Information Review, Vol. 33 Iss: 2, pp.329 – 342.
3. Brachman, Ronald J., Hector J. Levesque, and Raymond Reiter. "Knowledge representation and Reasoning". MIT press, 1992.
4. Grigoris Antoniou, Frank van Harmelen, "A semantic web primer", The MIT Press, 2004.
5. Stephen Davies, "Still building the memex", Communications of the ACM, v.54 n.2, 2011.
6. Knowledge representation and networked schemes [Tiešsaiste] Pieejams: [http://stpk.cs.rtu.lv/sites/all/files/stpk/lecture\\_7.pdf](http://stpk.cs.rtu.lv/sites/all/files/stpk/lecture_7.pdf) (27.05.2016.)
7. Zhang, Lei. "Knowledge graph theory and structural parsing." Twente University Press, 2002.
8. RDF 1.1 Primer [Tiešsaiste] Pieejams: <https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/> (27.05.2016.)
9. Pujara, Jay, Hui Miao, Lise Getoor, and William Cohen, "Knowledge graph identification." The Semantic Web–ISWC 2013. Springer Berlin Heidelberg, 2013. 542-557.
10. What is a Graph Database [Tiešsaiste] Pieejams: <http://neo4j.com/developer/graph-database/> (27.05.2016.)
11. Eseryel, Deniz, U. Yeliz Eseryel, and G. S. Edmonds. "Knowledge management and knowledge management systems." Intelligent learning infrastructures for knowledge intensive organizations: A semantic web perspective (2005): 105-145.
12. Cheong, Ricky KF, and Eric Tsui. "The roles and values of personal knowledge management: an exploratory study." Vine 40.2 (2010): 204-227.
13. Vicknair, Chad, et al. "A comparison of a graph database and a relational database: a data provenance perspective." Proceedings of the 48th annual Southeast regional conference. ACM, 2010.

14. Godwin-Jones, Robert. "Emerging technologies: blogs and wikis: environments for on-line collaboration.: An article from: Language, Learning & Technology." (2006).
15. Liana Razmerita, Kathrin Kirchner, Frantisek Sudzina, *Personal Knowledge Management: The role of Web 2.0 tools for managing knowledge at individual and organisational levels*, Online Information Review , vol. 33, no. 6, pp. 1021-1039, 2009.
16. Google Drive [Tiešsaiste] Pieejams: <https://www.google.com/drive> (29.05.2016.)
17. Evernote [Tiešsaiste] Pieejams: <https://evernote.com> (29.05.2016.)
18. OneNote [Tiešsaiste] Pieejams: <https://www.onenote.com/> (29.05.2016.)
19. DEVONthink [Tiešsaiste] Pieejams:  
<http://www.devontechnologies.com/products.html> (29.05.2016.)
20. Neo4j datubāze [Tiešsaiste] Pieejams: <http://neo4j.com/> (29.05.2016.)
21. jQuery [Tiešsaiste] Pieejams: <https://jquery.com/> (29.05.2016.)
22. Cypher language [Tiešsaiste] Pieejams: <http://neo4j.com/developer/cypher-query-language/> (29.05.2016.)

## Pielikums 1. Darbības ar neo4j datubāzi

```
function saveBlogEntry(){
    var query = {"query":"CREATE (:Blog {
        date: '"+new Date()+"', title:'"+$blogTitle.val()+"',
        text:'"+$blogTxt.val()+" })", "params" : {}};
    $.ajax({
        type:"POST",
        url: "http://localhost:7474/db/data/cypher",
        data: query,
        success: function(data, textStatus, jqXHR){
            getBlogs();
        },
        error: function(jqXHR, textStatus, errorThrown){
            showError(textStatus, errorThrown);
        }
    });
};

function deleteEntry(){
    var query = {"query":"MATCH (n) WHERE ID(n)='"+$(this).attr("data-id")
+" OPTIONAL MATCH (n)-[r]-() OPTIONAL MATCH (-[r2]-)(n) DELETE n, r, r2"};
    $.ajax({
        type:"POST",
        url: "http://localhost:7474/db/data/cypher",
        dataType: "json",
        data: query,
        success: function(data, textStatus, jqXHR){
            getBlogs();
        },
        error: function(jqXHR, textStatus, errorThrown){
            showError(textStatus, errorThrown);
        }
    });
};
```

```

function connectData(){
    var query = {"query":"MATCH (n),(s) WHERE ID(n)="
    +$(this).attr("data-id")
    +" AND ID(s)="+$selectedEvent.id
    +" CREATE (n)-[r:RELATED_TO]->(s) RETURN r"};
    $.ajax({
        type:"POST",
        url: "http://localhost:7474/db/data/cypher",
        dataType: "json",
        data: query,
        success: function(data, textStatus, jqXHR){
            getBlogs();
        },
        error: function(jqXHR, textStatus, errorThrown){
            showError(textStatus, errorThrown);
        }
    });
};

```

Bakalaura darbs „Personīgās zināšanu sistēmas” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Svetlana Poļakova

Rekomendēju/nerekomendēju darbu aizstāvēšanai

Vadītājs: asociētais profesors Dr. dat. Uldis Straujums

Recenzents:

Darbs iesniegts Datorikas fakultātē 30.05.2016.

Dekāna pilnvarotā persona: metodiķe Ārija Sproģe

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

\_\_\_\_\_

Komisijas sekretārs: