

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Geir Tagapere 142763IAPB

**AJASTATUD ÕPPIMISE ALGORITMI
RAKENDAMINE RELVALOA
TESTIMISKESKKONNALE**

Bakalaureusetöö

Juhendaja: Martin Verrev

MSc

Tallinn 2018

Autorideklaratsioon

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

Autor: Geir Tagapere

23.05.2018

Annotatsioon

Käesoleva lõputöö eesmärgiks on arendada kaks sarnast relvaloa taotlejatele mõeldud e-õppe keskkonda, millest ühele neist on rakendatud ajastatud õppimise (*spaced learning*) algoritm ja teisele mitte. Tulemuste võrdlemiseks ja analüüsimiseks salvestatakse testidele vastajate tulemused andmebaasi.

Töö esimeses pooles analüüsitakse olemasolevaid lahendusi ja kirjeldatakse nõuded loodud veebirakendusele. Töö teises pooles kirjeldatakse loodud veebirakendust, antakse ülevaade arengusuundade kohta ning tehakse kokkuvõte.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 8 peatükki, 19 joonist, 3 tabelit.

Abstract

Implementation of spaced learning algorithm on tests for acquiring firearms licence

The aim of the thesis is to develop two similar testing environments where one version of it has implemented spaced learning algorithm and the other has not. The application is based on tests created for mastering the Weapons Act. In order to compare the efficiency of both applications, respondents test results are saved in the database.

The thesis consists of comparison of other existing applications, describing the requirements for developed application and describing the built application.

The application is built in Java using Spring Boot and follows MVC architecture blueprint.

The thesis is in Estonian and contains 29 pages of text, 8 chapters, 19 figures, 3 tables.

Lühendite ja mõistete sõnastik

CSRF	<i>Cross Site Request Forgery</i> , päringuvõltsing
JPA	Java Persistence API
ORM	<i>Object-relational mapping</i> , objekt-relatsiooniline kaardistamine
MVC	<i>Model-View-Controller</i> , tarkvara arhitektuurimuster
RESTful	<i>Representational State Transfer</i> , tarkvaraarhitektuuri stiil
Default	Vaikimisi valik
Clickjacking	Klõpsurööv
Template engine	Šabloonimootor
Boilerplate kood	kood või koodilõik, mis ilmneb korduvalt
Session fixation	valideeritud kasutaja sessioonivõtme vargus
Standalone rakendus	iseseisev rakendus, mis jookseb arvutis eraldi protsessina
HTML	<i>HyperText Markup Language</i> , keel veebilehtede kuvamiseks
POST meetod	Päring andmete andmebaasi saatmiseks
GET meetod	Päring andmebaasist andmete saamiseks
IDE	<i>Integrated development environment</i> , integreeritud arenduskeskkond

Sisukord

1 Sissejuhatus	10
2 Taust	11
2.1 Mõisted	11
2.2 Õpitud materjali meeldejätmise	11
2.3 Enamlevinud, SuperMemo2, ajastatud õppimise algoritmi kirjeldus.....	13
2.4 Olemasolevad rakendused	14
2.4.1 Anki	14
2.4.2 The Mnemosyne Project.....	15
2.4.3 Synap	16
2.4.4 DuoLingo.....	17
2.4.5 Quizlet	18
2.4.6 SuperMemo	18
2.4.7 Fresh Memory	19
2.4.8 OpenCards	20
2.5 Olemasolevate rakenduste analüüs	20
2.6 Lõputöös kasutatav algoritm	22
3 Nõuded rakendusele	23
3.1 Funktsionaalsed nõuded	23
3.2 Mittefunktsionaalsed nõuded.....	23
4 Kasutatavad tehnoloogiad ja vahendid	25
4.1 Spring boot (1.5.13 SNAPSHOT)	25
4.1.1 Spring Web.....	25

4.1.2 Spring Security	25
4.2 Lombok.....	26
4.3 JPA.....	26
4.4 Thymeleaf.....	26
4.5 MariaDB	27
5 Realisatsioon.....	28
5.1 Domeenimudel.....	28
5.2 Küsimuste salvestamine andmebaasi.....	29
5.3 HTML lehed vaadetega	30
5.3.1 login.html.....	31
5.3.2 home.html	31
5.3.3 traffictests.html	32
5.3.4 firearmsTesting.html	32
5.3.5 results.html	33
6 Valideerimine	35
7 Võimalikud arengusuunad.....	37
7.1 Küsimuste hulga muutmine	37
7.2 Sisselogimise valikuvariandid	37
7.3 Muud testid.....	37
7.4 Efektivsem algoritm	38
8 Kokkuvõte	39
Kasutatud kirjandus	40
Lisa 1	43
Lisa 2	44

Jooniste loetelu

Joonis 1. Unustamiskurv	12
Joonis 2. Harilik ajastatud õppimise tõhususe kontrollimise protsess	12
Joonis 3. Anki õpikaartide kuvamine	14
Joonis 4. Mnemosyne õpikaartide kuvamine	16
Joonis 5. Leitneri süsteem.	17
Joonis 6. DuoLingo näide sõnade ilmnemise sagedusest	17
Joonis 7. SuperMemo unustamiskurv.....	18
Joonis 8. Lomboki "@Getter" annotatsiooni funktsionaalsus.....	26
Joonis 9. Domeenimudel	29
Joonis 10. Küsimuse salvestamine andmebaasi	30
Joonis 11. Navigeerimisriba väljalogituna	31
Joonis 12. Navigeerimisriba sisselogituna	31
Joonis 13. <i>login.html</i> vaade	31
Joonis 14. <i>Home.html</i> vaade.....	31
Joonis 15. <i>trafficTests.html</i> tagastamine.....	32
Joonis 16. <i>traffictests.html</i> vaade.....	32
Joonis 17. <i>firearmsTesting.html</i> vaade.	33
Joonis 18. <i>results.html</i> esitatud vastuste vaade.....	33
Joonis 19. <i>results.html</i> kõikide tulemuste vaade	34

Tabelite loetelu

Tabel 1. Olemasolevate rakenduste võrdlus	21
Tabel 2. Tuupimise taktikaga saadud tulemused.....	35
Tabel 3. Ajastatud õppimise taktikaga saadud tulemused.....	35

1 Sissejuhatus

Inimesed kipuvad õppimise jätma viimasele minutile, püüdes korraga liiga palju informatsiooni meelde jätta. Seetõttu järgneb harilikult peale sündmust, milleks õpiti, õpitu unustamiskurvile (*forgetting curve*) [1] omane eksponentsiaalne unustamine. Et omandatud teadmised püsiks meeles ka edaspidi, on efektiivne seda teostada viisil, mida kirjeldas Hermann Ebbinghaus 1885. aastal enda hüpoteesis. Nimelt soodustab õpitu meelepidamist ajastatud õppimine (*spaced learning*) [2].

Antud töö eesmärgiks on tõsta õpiefektiivsust, kasutades selleks ajastatud õppimise meetodit. Selle saavutamiseks on autor loonud rakenduse, mis võimaldab harjutada relvaeksami teooriaosa. Selleks, et õppimist efektiivsemaks muuta:

- Kirjeldan algoritmi, millega saavutatakse spaced learning õpiviis
- Realiseerin selle olemasolevas rakenduses
- Mõõdan ja võrdlen respondentide tulemusi tavalise ja algoritmilise rakenduse vahel

Töö tulemusena valmib ajastatud õppimise algoritmiga veebirakendus, kuhu on edaspidi laienemise mõttes võimalik rakendada teisi selliseid teste, mille sisu sõltub seadustest, näiteks autokooli või väikelaevajahi loa testid. Loodava veebikeskkonna eesmärk on olla hariv ja seda viisil, et õpitu säiliks mälus maksimaalselt pikalt, sest eelnimetatud luba(sid) omades on oluline ohutusnõuete ja seaduse tundmine. Samuti selgub töö tulemusena kasutatava algoritmi efektiivsus.

2 Taust

Selles peatükis kirjeldab autor käsitletava teema paremaks mõistmiseks lahti mõisted, analüüsib olemasolevaid rakendusi ning kirjeldab töös kasutatavat algoritmi.

2.1 Mõisted

Lühimälu (*short-term memory*) – Mäluliik, millele on omane lühike kestus ja väike maht, millest vaid väike osa säilib püsivamates mäluliikides [27]. Tavaliselt salvestame lühimällu näiteks takso teenust pakkuva taksojuhi nime.

Unustamiskurv (*forgetting curve*)– Kurv, mis kirjeldab informatsiooni unustamist läbi aja [26].

Ajastatud õppimine (*spaced learning*) – Õpimeetod, mis praktiseerib vahede jätmist lühikeste õpissessioonide vahele, eesmärgiks salvestada materjal pikaajalisse ehk semantilisse mällu.

Tuupimine (*cramming*) – Õpitu, tavaliselt faktide, meeldejätmise seoseid loomata.

2.2 Õpitud materjali meeldejätmise

Inimene õpib terve elu. Enamus õpitust ununeb materjali kordamata kiiresti. Unustamise kiirust kirjeldab tänapäeval valem [22]:

$$R = e^{-t/S},$$

kus R – info mälust kättesaadavuse lihtsus,

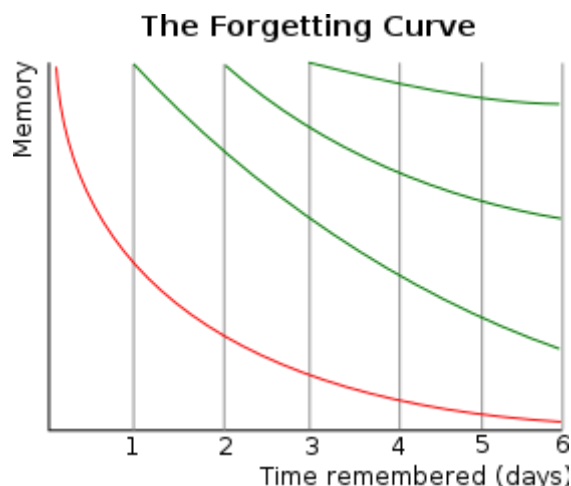
S – mälu stabiilsus. Määrab, kui kiirelt R aja jooksul puuduliku kordamisega langeb,

t – aeg.

Üheks unustamise põhjuseks on õpitava materjali tuupimine (*cramming*), mis tähendab, et õpitu talletatakse lühimällu. Unustamiskurvi languse minimeerimiseks on autor

loonud relvaloa taotlejatele kaks veebirakendust, millest ühes on rakendatud ajastatud õppimise põhimõtetega lihtsustatud algoritm ja teises mitte. Alloleval joonisel demonstreeritakse unustamiskurvi [25].

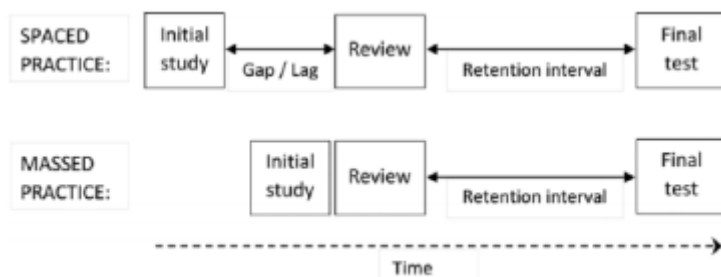
Joonis 1. Unustamiskurv



Pikaajalise meeldejätmise tõhususe tõstmiseks on soovituslik õppematerjali perioodiliselt korrata. Ajavahemik, mis peaks jääma kordamiste vahele, on varieeruv ja sõltub nii meeldejäetava materjali sisust, mis aja jooksul omandati, kuid ka vahemikust mis jääb momendi ja testimise vahele. Näiteks paluti ühes uuringus õpilastel meelde jätta „Ateena Vanne“ (*The Athenian Oath*) [21]. Esimene grupp õpilasi kuulis vannet 6 korda järjest, teine grupp aga 3 korda ühel päeval ja veel 3 korda 3 päeva hiljem. Esimesel päeval peale vande ettelugemisi toimus testimine mõlemale grupile. Tulemused näitasid, et esimesel grupil olid paremad sooritused. Järgmine testimine toimus 4 nädala pärast. Selles testis ületasid ajastatud õppijate sooritused selgesti esimese grupi omad. Tulemused näitasid, et ajastatud õppimine aitab kaasa õpitu parema meenutamisele [28].

Nii näeb tavaliselt välja ajastatud õppimise tõhususe kontrollimise protsess [28].

Joonis 2. Harilik ajastatud õppimise tõhususe kontrollimise protsess



2.3 Enamlevinud, SuperMemo2, ajastatud õppimise algoritmi

kirjeldus

SuperMemo2 algoritmi kasutati esmakordselt SuperMemo 2.0 rakenduses, mis loodi aastatel 1987-1989 [14]. Algoritmis kasutatakse iga objekti, rakenduses õpikaardi, uuesti ilmnemise arvutamiseks valemit, mille üheks sisendiks on kasutaja hinnang 0-5 süsteemis õpikaardi vastuse meeldetuletamise lihtsusele.

SuperMemo2 algoritmi rakendamine õpikaartidele:

1. Iga õpikaardi lisamisel andmebaasi määratakse sellele lihtsuse faktor (EF) väärtusega 2,5. Õpikaardile vastates varieerub selle väärtus vahemikus 1,3-2,5.
2. Õpikaarte kuvatakse vastavalt järgnevatele intervallidele (päevades)
 $I(1) = 1$
 $I(2) = 6$
 $n > 2$ puhul $I(n) = I(n-1)*EF$,
kus $I(n)$ – intervall päevades peale n-ndat kordamist; EF – E-Faktor. Lihtsuse faktor, mis presenteerib meeldejätmise ja meeldetuletamise lihtsust mälust. Murdarvu puhul ümardada lähima täisarvuni.
3. Õpikaardi vastust kuvades annab kasutaja vastuse meenutamise lihtsusele hinnangu skaalal 0-5, kus 0 tähendab üldse mitte mäletamist ning 5 mõtlemata vastuse teadmist.
4. Peale igat õpikaardi kordust muudetakse lihtsuse faktorit vastavalt valemile:
 $EF' = EF + (0,1 - (5-q) * (0,08 + (5-q) * 0,02))$,
kus EF' – uus E-faktori väärtus; EF – E-faktori vana väärtus; q – lihtsuse hinnang skaalal 0-5. Juhul, kui EF on väiksem kui 1,3, siis $EF = 1,3$.
5. Kui meenutamise lihtsusele antud hinnang oli väiksem kui 3, siis E-faktorit ei muudeta ning kaarti kuvatakse nagu see oleks alles lisatud ja küsitakse uuesti.

SuperMemo2 edasiareng SuperMemo1 suhtes seisneb E-faktori vahemiku muutmises. Vanem versioon seadis EF vahemikuks 1,1-2,5. E-faktori langedes alla 1,3 hakkas objekt häirivalt tihedalt ilmnema.

2.4 Olemasolevad rakendused

Ajastatud õppimist rakendavad mitmed tarkvarad, mida on saadaval nii tasuta kui tasulisi versioone. Järgnevalt kirjeldatakse mõningaid selliseid rakendusi.

2.4.1 Anki

Anki on põhjalikult dokumenteeritud, vaba lähtekoodiga õppimiskeskond, kuhu on rakendatud iseseadistatav *spaced learning* algoritm. Rakendus implementeerib modifitseeritud SuperMemo2 algoritmi.

Anki olemus seisneb küsimus-vastus liiki õpikaartidele vastamises. See tähendab, et kasutajal on võimalik ise luua erinevaid kaardipakke, kuhu saab oma soovi järgi lisada nii palju õpikaarte kui soovib ning vastavalt sellele, mida õppida soovib. Kaartidele on võimalik parema meelde jätmise huvides lisada nii pilte kui ka helifaile. Küsimuste sorteerimise eesmärgil saab kaartidele lisada märksõnu.

Testimisel saab ilmnevale küsimusele mõtteliselt vastata ning seejärel peale vastuse kuvamist valida, kuidas läks. Valikuteks on “Uuesti”, “Hästi” ning “Lihtne” [10]. Seadistused võimaldavad paika seada küsimuse järgmise ilmumise peale küsimusele vastamist. Vaikimisi on valiku “Uuesti” valimisel seatud küsimus ilmema uuesti samas õpiintervallis. Valik “Hästi” annab Ankile teada, et kasutaja teab õpikaardi vastest, kuid küsib seda peagi uuesti. Valides “Lihtne” ilmneb kaart 3 päeva pärast.

Joonis 3. Anki õpikaartide kuvamine



Olles kaardi varasemalt ära õppinud, saab kasutaja hakata küsimusi kordama. Tekib juurde valik “Raske” [11]. Teised valikud säilitavad vastamisel sama põhimõtte, kuid

väikeste erinevustega. Tehes valiku “Hästi”, küsitakse *default* sätete järgi vastavat küsimust uuesti umbes 2.5x aja pärast, mis oli sel hetkel kulunud küsimuse uuesti ilmnemiseks. “Lihtne” lükkab küsimust kaardipakis veelgi kaugemale. Vaikimisi sätete järgi tähendaks see “Lihtne” kaardi ilmnemist 1.3x pikema aja pärast kui “Hästi”.

Anki eeliseks on lihtne sünkroniseerimine erinevate seadmete vahel. Samuti lubab üle 100 000 õpikaardi probleemideta käsitlemist [12]. Ideaalne õpivahend lihtsate faktide meelde jätmiseks, kuid mitte informatsiooni õppimiseks, mis nõuab arusaamist keeruliste seoste vahel.

Kuigi Anki on põhjalikult dokumenteeritud, on selle miinuseks õpikaartide ilmnemise muutmist võimaldav süsteem. Olenemata sellest, et sätete redigeerimine pole kohustuslik, võib see võhikule arusaamatusi tekitada.

Hind: Windows, Mac, Linux, Android tasuta; iOS \$24.99

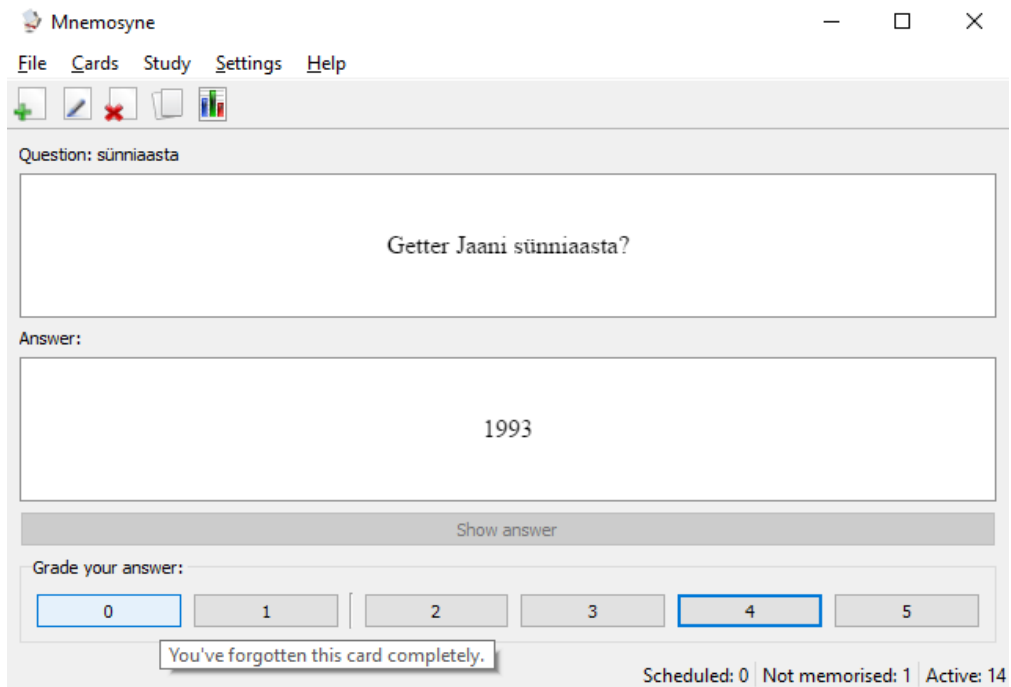
Platvormid: Windows, Mac OS, Linux/BSD, iPhone/Android

2.4.2 The Mnemosyne Project

Mnemosyne on põhjaliku dokumentatsiooniga, selgesti arusaadava ja lihtsasti kasutatava kasutajaliidesega õpikaartidega õppimise rakendus. Testimisel kuvatakse õpikaardi vastus kasutaja soovil. Seejärel võimaldatakse hinnata mälu 0-5 süsteemis.

- 0 – kasutaja ei mäletanud õpikaarti üldse
- 1 – kasutaja on vastuse unustanud
- 2 – kasutaja polnud vastuses päris kindel, küsimuse ilmnemise intervall oli liiga suur
- 3 – kasutaja vastas õigesti, kuid nõudis pingutust, küsimuse ilmnemise intervall võiks olla väiksem
- 4 – kasutaja vastas õigesti väikese viivitusega, küsimuse ilmnemise intervall peaks olema sobilik
- 5 - kasutaja vastas pingutusteta õigesti, küsimuse ilmnemise intervall võiks olla kiirem

Joonis 4. Mnemosyne õpikaartide kuvamine



Õpikaardid võivad sisaldada pilte, helisid, videosid, Flash animatsioone [13].
Sorteerimiseks on võimalik kaartidele lisada märksõnu.

Mnemosyne kasutab SuperMemo2 algoritmi, millele on tehtud väikseid muudatusi seoses õpikaartide intervallidega [17].

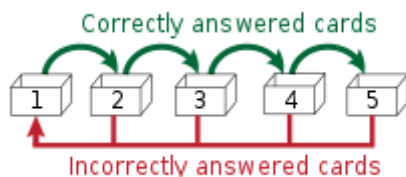
Hind: Tasuta

Platvormid: Windows, Mac OS, Linux, Android

2.4.3 Synap

Synap on Leitneri süsteemil [16] põhinev õpikeskkond. Tasuta Synap lubab kasutajal kord nädalas ajastatud õppimisega testi lahendada. Tasuline seevastu iga päev. Samuti kaasneb tasulise versiooniga piiramatute testide loomise võimalus, testidesse piltide lisamise võimalus, areneva soorituste jälgimise süsteem ja graafikud, võrguühenduseta lahendamine iOS appis. Synapis saab luua ainult küsimusi, millel on valikvastused. Miinuseks on nõue, mis kohustab inimest valikutesse sisestama vähemalt üht vale vastust.

Joonis 5. Leitneri süsteem. „In the Leitner system, correctly answered cards are advanced to the next, less frequent box, while incorrectly answered cards return to the first box.“



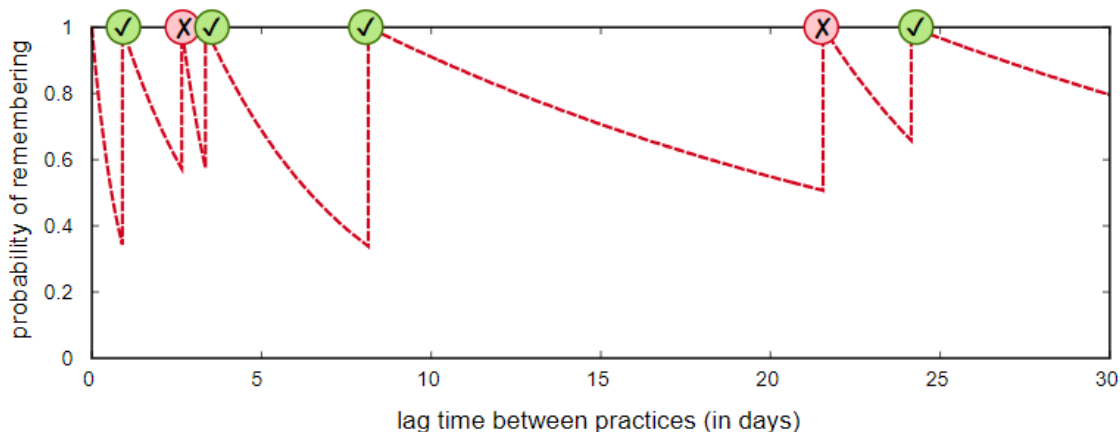
Hind: Piiratud funktsionaalsusega on tasuta. Tasuline £2.99 kuu.

Platvormid: Veeb, Android, iOS

2.4.4 DuoLingo

DuoLingo on innovaatiline keeleõppimise keskkond, mis pakub ingliskeelsele õppele 31 erinevat keelevalikut. Õppimine toimub lugedes, kirjutades, kuulates ning rääkides. DuoLingo kasutab sõna õppimiseks nende enda poolt loodud süsteemi *Half-life regression* [23], mis üritab varasemate vastuste põhjal ära arvata, millal kasutaja mälutugevus mingi sõna suhtes on umbes 50% ning siis kuvab selle. Alloleval joonisel on DuoLingo näidisõnade värskendamise sagedusest. Punane ristike tähistab valesti vastamist ja roheline tähistab õigesti vastamist.

Joonis 6. DuoLingo näide sõnade ilmnemise sagedusest



DuoLingole rakendatud Half-life regression süsteem parandab sõna ilmnemise viga varasema Leitneri süsteemiga võrreldes umbes 24% pealt ligikaudu 13% protsendile [23].

Hind: Tasuta. Tasuline versioon eemaldab reklaamid.

Platvormid: Veeb, Android, iOS

2.4.5 Quizlet

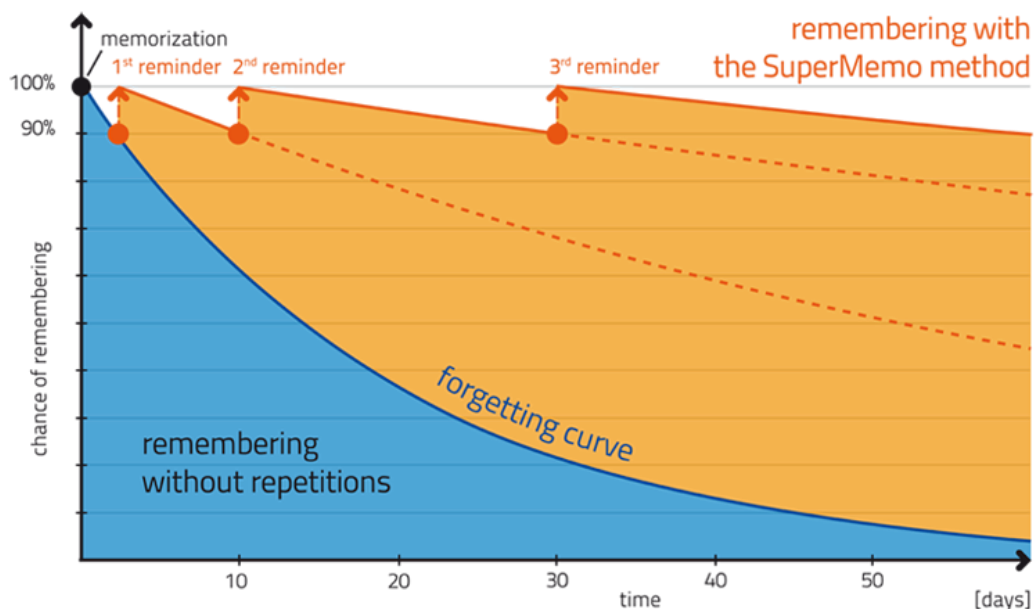
Quizlet on testimiskeskond, kus kasutajal on võimalik ise teste koostada, kuid ka lahendada valmisolevaid. Valmisolevad testid on loodud Quizleti keskkonna kasutajate poolt. Nende miinuseks on ebakindlus tõepärasuses. Teste on võimalik lahendada õpikaartide stiilis, küsimus-vastus stiilis, õige või vale stiilis, paaride sobitamisega, aja peale vastates ning valikvastustega. Tasulises versioonis saab küsimustele lisada nii pilte kui ka audiot. Kasutaja saab kuvada enda testide tulemuste statistikat. Tasulises, *spaced learning* rakenduses, on võimalik lubada teadete kuvamine, et testimine meelde tuletada.

Hind: Piiratud funktsionaalsusega tasuta. Spaced learning on rakendatud tasulisse versiooni, 1.67€ kuu.

Platvormid: Veeb, Android, iOS

2.4.6 SuperMemo

Joonis 7. SuperMemo unustamiskurv



SuperMemo oli esimene maailmas, kes kasutas arvuteid intervallide optimeerimiseks õppimise korduste vahel [18]. Praeguseks on kasutusel SuperMemo17 algoritm. Võimalik

on lahendada kasutaja loodud ja olemasolevaid teste. Kasutaja loodud testidesse on võimalik lisada pilte ja audio. Osad olemasolevad testid on tasulised. Testid ja tulemused on seadete vahel sünkroniseeritud.

Peale küsimuse kuvamist saab kasutaja küsida SuperMemolt vastuse nägemist. Seejärel tekib kasutajal kolm valikut [19].

[19] <https://www.supermemo.com/en/blog/repeat-section-and-3-buttons-assessment>

- „*I don't know*“ – Küsimuse intervall muutub kiiremaks. Samuti kuvatakse küsitud küsimus enne testi lõppemist uuesti, et kasutajale vastus kinnistuks.
- „*Almost*“ – Küsimuse intervall muutub kiiremaks, kuid säilib võimalus, et vastav küsimus esitatakse samal kordamisel uuesti.
- „*I know*“ – Küsimuse intervall pikeneb. Pikkus sõltub varem antud vastustest.

Hind: Tasuta õpikaartide loomise võimalus piiratud mahus. Erinevad kursuste komplektid maksavad.

Platvormid: Veeb, Windows, Android, iOS

2.4.7 Fresh Memory

Fresh Memory kasutab ajastatud õppimiseks SuperMemo2 algoritmi. Ainukeseks õpivahendiks on õpikaardid, kuhu on lisaks tekstile võimaldatud piltide lisamine [20]. Kasutaja saab peale küsimusele mõttelist vastamist hinnata oma mälu 1-5 hindelises süsteemis.

- 1 (*Unknown*) – Küsimus täielikult ununenud, küsitakse samas testis uuesti.
- 2 (*Incorrect*) – Küsimusele vastati valesti, küsitakse samas testis uuesti.
- 3 (*Difficult*) – Küsimusele oli raske vastata, viimane intervall oli liiga pikk.
- 4 (*Good/OK*) – Küsimuse vastus meenub aegsasti, intervall oli paras.
- 5 (*Easy*) – Küsimus oli liiga lihtne, intervall oli liiga lühike.

Küsimuste otsa saamise korral saab kasutaja huvi korral segaesituses edasi vastata ilma, et see sisaldaks ajastatud õppimist. Küsimuste komplekt on võimalik Fresh Memory programmi importida. Küsimuste kasutajapõhine informatsioon salvestatakse eraldi faili. Ilma viimase failita arvestatakse küsimusi kui uusi küsimusi.

Hind: Piiratud funktsionaalsusega on tasuta. Tasuline 9\$.

Platvormid: Windows, Linux

2.4.8 OpenCards

OpenCards on vaba lähtekoodiga ajastatud õppimise eesmärgil loodud rakendus, mis kasutab OpenCards poolt täiustatud SuperMemo2 algoritmi. OpenCards rakendusse on võimalik importida õpikaarte .ppt laiendusega failist [15]. Õpikaartidele saab lisada audio, pilte, graafikuid, valemeid. Valida saab kahe meetodi vahel. Esimene neist on „*LastMinute*“. See võimaldab õpikaardid läbida kolmel moel:

1. Eksisteerib kindel kogus kastikesi. Õigesti vastates liigub kaart ühe võrra paremale. Valesti vastates liigub ühe võrra vasakule või kui kaart on esimeses kastis, püsib paigal. Vastamine õpikaartidele, kuni kõik kaardid on parempoolses kastis.
2. Lõpeta testimine, kui sätestatud ajalimiit on täis.
3. Lõpeta testimine, kui sätestatud kogusele kaartidele on vastatud.

Teine on „*Start Learning*“ ehk ajastatud õppimisega algoritmitud meetod.

Sünkroonimist teiste seadmetega ei toimu, kuigi sünkroonida on võimalik importides kasutajapõhine faili, milles on *spaced learning* algoritmiliseks vastamiseks vajalikud andmed.

Hind: Tasuta

Platvormid: Windows, Linux, Mac

2.5 Olemasolevate rakenduste analüüs

Käesolevas alapeatükis võrreldakse kaheksa võrdlusesse võetud ajastatud õppimise algoritmiga tarkvara või veebirakenduse iseärasusi.

SuperMemo2 algoritm on lihtne, kuid efektiivne. Kokku on SuperMemo algoritme seitseteist. Selles lõputöös kasutavad uuritud tarkvaradest SuperMemo2 ajastatud õppimise algoritmi kaheksast rakendusest viis (Tabel 1). Mõnede tarkvarade või

veebirakenduste loojad on omalt poolt SuperMemo2 algoritmi täiendanud. Kahel rakendusel pole algoritmi nimetatud, kuid on teada, et kasutavad ajastatud õppimist.

Kõik rakendused on tasuta. Mõned rakendused pakuvad raha eest lisafunktsionaalsust. Seadmepõhiselt on maksustatud ainult Anki iOS rakendus. Quizlet on kuutasu maksmata ajastatud õppimise mõistes kasutu, sest ajastatud õppimine rakendatakse tasulises versioonis.

Tabel 1. Olemasolevate rakenduste võrdlus

Nimi	Platvormid	Litsents	Hind	Algoritm
Anki	Windows, Mac OS, Linux/BSD, iPhone/Android	GNU AGPL v3	Windows, Mac, Linux, Android tasuta; iOS \$24.99	Modifitseeritud SuperMemo2
The Mnemosyne Project	Windows, Mac OS, Linux, Android	GNU GPL v2	Tasuta	Modifitseeritud SuperMemo2
Synap	Veeb, Android, iOS	Edasiantav litsents	Piiratud funktsionaalsusega on tasuta. Tasuline £2.99 kuu	Leitneri süsteem
DuoLingo	Veeb, Android, iOS	All-litsents	Tasuta, tasuline versioon eemaldab reklaamid	<i>Half-life regression</i>
Quizlet	Veeb, Android, iOS	All-litsents	Piiratud funktsionaalsusega tasuta. Spaced learning on rakendatud tasulisse	Spaced repetition (teadmata)

			versiooni, 1.67€ kuu.	
SuperMemo	Veeb, Windows, Android, iOS	Omanduslik tarkvara	Tasuta õpikaartide loomise võimalus piiratud mahus. Erinevad kursuste komplektid maksavad.	SuperMemo17
Fresh Memory	Windows, Linux	GNU GPL v3	Piiratud funktsionaalsusega on tasuta. Tasuline 9\$.	SuperMemo2
OpenCards	Windows, Linux, Mac	BSD	Tasuta	Modifitseeritud SuperMemo2

2.6 Lõputöös kasutatav algoritm

Lõputöös kasutab autor arvutusressursi kokkuhoiu ja testimise eesmärgil SuperMemo2 algoritmist veelgi lihtsamat algoritmi. Algoritmi idee on tuletatud aine „Programmeerimise põhikursus Javas (2015)“ esimesest kodutööst, milleks oli tikumäng, lisäülesandest „võidukäigu õppimine“ [24].

Algoritm leiab kõigepealt kõik ühte liiki testi, antud juhul relvatesti, küsimused ning annab neile kaaluks 1. Seejärel leitakse kasutaja kõik selle testitüübi lahendatud testid ajaliselt järjestatult ja kontrollitakse, millistele küsimustele on õigesti vastatud ja millistele valesti. Õige küsimuse puhul eemaldatakse kaalult 1, kuid kaal alla 1 minna ei saa. Vale küsimuse puhul lisatakse kaalule 1 juurde. Kõikidest küsimustest leitakse kümme suvalist küsimust. Suurema kaaluga küsimuse esinemise tõenäosus on teistest kõrgem. Nii kuvatakse kõige enam valesti vastatud küsimused suurema tõenäosusega. Algoritmilise rakenduse lähtekoodi leiab Bitbucketi repositooriumist (vt Lisa 2).

3 Nõuded rakendusele

Selles peatükis kirjeldab autor veebirakenduse funktsionaalseid ja mittefunktsionaalseid nõudeid. Funktsionaalsetest nõuetest lähtuvalt on realiseeritud lõputöö rakendus. Mittefunktsionaalsed nõuded kirjeldavad töökäiku.

3.1 Funktsionaalsed nõuded

Funktsionaalsed nõuded on koostatud vastavalt vajadustele, et realiseerida loodud ajastatud õppimise algoritmiga testimiskeskond.

Kirjeldame loodava veebirakenduse funktsionaalsed nõuded:

1. Respondent saab testimiskeskonda sisse logida
2. Respondendi kasutajakonto luuakse esmakordsel Facebookiga sisselogimisel
3. Autenditud kasutaja saab välja logida
4. Testimiseks peab kasutaja olema sisselogitud
5. Respondent vastab küsimustikule, mis koosneb n-arvust testküsimustest
6. Küsimustik koostatakse respondendile individuaalselt vastavalt varem vastatud õigetele-valedele vastustele
7. Respondendi vastused salvestatakse isikustatult, s.t. iga vastuse juures on teada selle vastaja ja vastamise kellaeg
8. Peale küsimustiku täitmist kuvatakse respondendile õiged ja valed vastused
9. Respondent saab kuvada enda varasemaid tulemusi
10. Käesolevas prototüübis ei ole kaetud administraatori vaadet – s.t. küsimused on etteantud ja loetakse andmebaasist

3.2 Mittefunktsionaalsed nõuded

Kirjeldame loodava veebirakenduse mittefunktsionaalsed nõuded:

1. Veebirakenduse kood peab olema ingliskeelne.
2. Veebirakendus kood on kirjutatud Java keeles.
3. Veebirakendus on eestikeelne.

4 Kasutatavad tehnoloogiad ja vahendid

4.1 Spring boot (1.5.13 SNAPSHOT)

Spring Boot [3] on Javas arendamiseks mõeldud MVC raamistik, mille üks fundamentaalne funktsionaalsus on sõltuvuste süstimine (*dependency injection*). Seetõttu on võimalik enamus olemasolevaid teeke ja tehnoloogiaid projekti kaasata. Suureks miinuseks on esialgne keeruline konfigureerimine, et üldse projekti IDE's (*Integrated development environment*) jooksutada saaks.

Spring boot on Spring raamistiku edasiarendatud, lihtsustatud sõltuvuste süstimise tööriist. Kuigi Spring framework võimaldab teha kõike sama, mida Spring bootki, on viimase eeliseks projektikomplekti loomise võimalikkus, millega kaasneb seadistatud konfiguratsioon. Aega aitab kokku hoida ka XML konfiguratsiooni

Spring boot võimaldab projekti kokku panna [4] ilma suurema vaevata, valides kodulehel sobilikud teegid ja sõltuvused, samuti aidates ära hoida *boilerplate* koodi.

4.1.1 Spring Web

Kuna autori loodud rakendus on veebirakendus, siis lisati projekti ka Spring Web *dependency*. Nimetatud sõltuvus lisab projekti omakorda teisigi sõltuvusi [7]. Java koodi jooksutamiseks on autori jaoks nendest tähtsaim Tomcati veebiserveri keskkond.

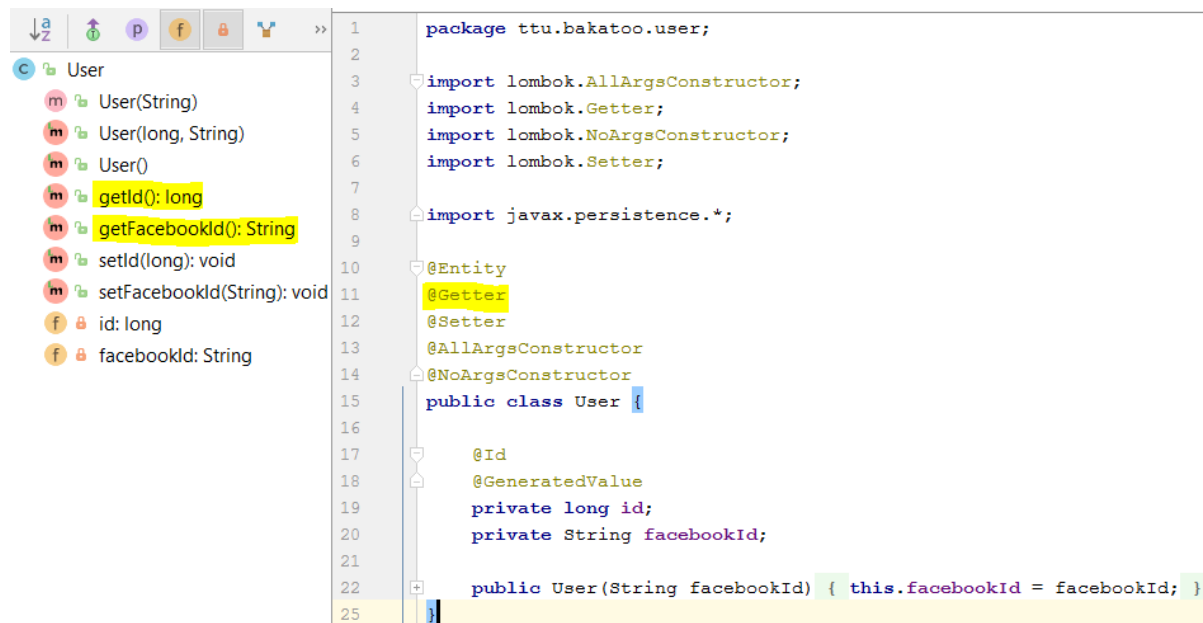
4.1.2 Spring Security

Spring Security *dependency* pakub Java rakendustele autentimist ja autoriseerimist. Selles lõputöös kasutatakse seda ainult autentimise eesmärgil. Spring Security tagab turvalisuse *session fixation*, *clickjacking* [9], CSRF vastu [6].

4.2 Lombok

Lombok *dependency* aitab projektis ridade arvelt kokku hoida ja vähendada *boilerplate* koodi. Kasutades klassis erinevaid Lomboki annotatsioone, genereerib Lombok vastavalt annotatsioonile vajalikud meetodid. Alloleval pildil on näha, kuidas näiteks `@Getter` konstrueerib `getId()` ning `getFacebookId()`.

Joonis 8. Lomboki "@Getter" annotatsiooni funktsionaalsus



```
1 package ttu.bakatoo.user;
2
3 import lombok.AllArgsConstructor;
4 import lombok.Getter;
5 import lombok.NoArgsConstructor;
6 import lombok.Setter;
7
8 import javax.persistence.*;
9
10 @Entity
11 @Getter
12 @Setter
13 @AllArgsConstructor
14 @NoArgsConstructor
15 public class User {
16
17     @Id
18     @GeneratedValue
19     private long id;
20     private String facebookId;
21
22     public User(String facebookId) { this.facebookId = facebookId; }
23
24 }
```

The screenshot shows an IDE with a project explorer on the left and a code editor on the right. The project explorer shows a package 'ttu.bakatoo.user' containing a class 'User'. The class 'User' has several methods listed: 'User(String)', 'User(long, String)', 'User()', 'getId(): long', 'getFacebookId(): String', 'setId(long): void', and 'setFacebookId(String): void'. The 'getId(): long' and 'getFacebookId(): String' methods are highlighted in yellow. The code editor shows the source code for the 'User' class, which includes imports for Lombok annotations and JPA annotations. The class is annotated with `@Entity`, `@Getter`, `@Setter`, `@AllArgsConstructor`, and `@NoArgsConstructor`. The class has two private fields: `private long id;` and `private String facebookId;`. There is a constructor `public User(String facebookId) { this.facebookId = facebookId; }`. The `@Getter` annotation is highlighted in yellow, and the generated methods `getId(): long` and `getFacebookId(): String` are also highlighted in yellow in the project explorer.

4.3 JPA

JPA on spetsifikatsioon, mis on kokku pandud erinevatest ORM API-dest. JPA aitab Java objekte relatsioonilisse andmebaasi tabelisse kaardistada ja vastupidi.

4.4 Thymeleaf

Autor kasutab Thymeleafi käesolevas töös vaadete renderdamiseks. Thymeleaf on kaasaegne Java *template engine*, mis pakub võimalust täielikuks Spring raamistiku integratsiooniks, kuid mida saab kasutada ka näiteks *standalone* rakendustes. Thymeleaf implementeerib naturaalse vaadete kontseptsiooni, mis tähendab, et vaateid saab avada otse browseris ilma serverit käivitamata [8].

4.5 MariaDB

Andmebaasi haldamiseks kasutab autor MariaDB, mis on üks enimkasutatud [5] täielikult vaba lähtekoodiga andmebaaside tarkvarasid maailmas. MariaDB on skaleeritav, relatsiooniline andmebaas, mis kasutab operatsioonide sooritamiseks SQL lausungeid.

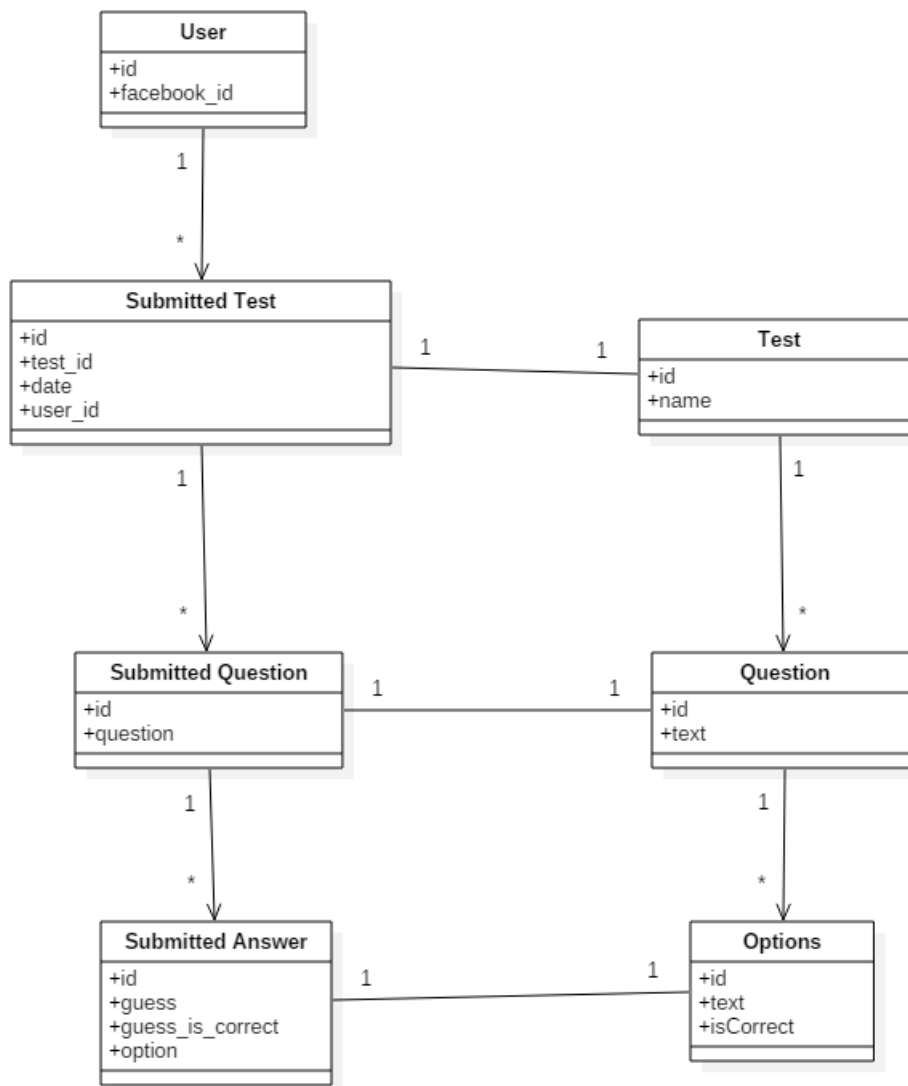
5 Realisatsioon

Autor kasutas projekti alge loomiseks Spring Initializr'it. Valinud <http://start.spring.io> lehel sobivad valikud ning sõltuvused, saab nupuvajutusega genereerida projekti. Käesolevas lõputöös valiti projektitüübiks Gradle, keeleks Java ning Spring Boot versiooniks 1.5.13 (SNAPSHOT). Sõltuvusteks valiti Web, Security, JPA, Lombok, Thymeleaf ja manuaalselt lisati juurde MariaDB, Facebook ning Thymeleaf Spring Security. Eelnimetatud sõltuvused on vajalikud rakenduse töötamiseks.

5.1 Domeenimudel

Lõputöö raames loodud rakenduse tööprotsessi kirjeldab joonis 10.

Joonis 9. Domeenimudel



5.2 Küsimuste salvestamine andmebaasi

Kasutajate testimise võimaldamiseks peavad küsimused olema eelnevalt andmebaasi salvestatud. Selleks tuleb kõigepealt luua valikvastus ehk siis *Options* objekt, mille väljadeks on *id* ja *text*, mis on ühe valikvariandi tekst ning *isCorrect*, mis ütleb kas tegemist on õige või vale vastusega. Valikvastuse objekt salvestatakse andmebaasi *options* tabelisse. Seejärel luuakse *Question* objekt, mille väljadeks on *id* ning *text*, millest viimane tähistab küsimuse teksti. Valikvastused kaardistatakse relatsioonilises tabelis küsimuste külge mõlema, nii valikvastuse kui küsimuse, tabeli *id* veeru abil. Küsimuse objekt salvestatakse andmebaasi *question* tabelisse. Viimaks luuakse *Test* objekt, mille väljaks on sõne *name*. *Name* on konstant enum tüüpi klassist *TestName*.

Küsimus kaardistatakse relatsioonilises tabelis testi külge mõlema, nii *question*, kui ka *test*, tabeli *id* veeru järgi. *Test* objekt salvestatakse andmebaasi tabelisse *test*.

Relvaloa testides kuvatavad küsimused luuakse ja salvestatakse *Bootstrap.java* failis.

Joonis 10. Küsimuse salvestamine andmebaasi

```
@Override
public void run(ApplicationArguments args) throws Exception {
    if (testRepository.findAll().size() == 0) {
        Options o1 = optionRepository.save(new Options("Reks", true));
        Options o2 = optionRepository.save(new Options("Pontu", false));
        Options o3 = optionRepository.save(new Options("Pluto", true));
        Options o4 = optionRepository.save(new Options("Doge", false));

        Question q1 = new Question("Koera nimi?", Arrays.asList(o1,o2,o3,o4));
        questionRepository.save(q1);
        Test test1 = new Test(TestName.RELVAEKSAM, Arrays.asList(q1));
        testRepository.save(test1);
    }
}
```

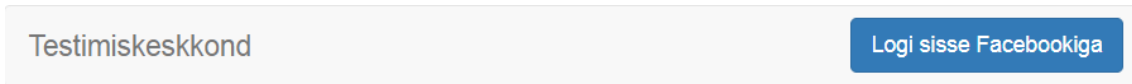
5.3 HTML lehed vaadetega

Selleks, et rakenduse tööd visuaalselt kuvada, tagastab Thymeleaf lõputöö rakenduses viite erinevasse HTML lehte kokku kaheksa vaadet. Veebirakendus HTML lehtedeks on:

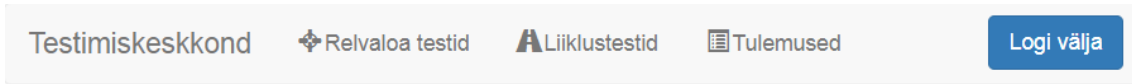
1. login.html
2. home.html
3. trafficTests.html
4. fireArmstesting.html
5. results.html

Igasse HTML lehte on sisaldatud Thymeleaf abil *header.html* fail, mis kuvab lehe navigeerimisriba. Kasutaja esmakordsel sisenemisel keskkonda registreeritakse tema kasutajakonto andmebaasi. Kasutaja tuvastatakse Facebook ID kaudu. Navigeerimisriba elementide tinglikku kuvamist saavutatakse Spring Security väljenditega *isAnonymous()* ja *isAuthenticated()*. Joonisel 10 ja joonisel 11 on näidatud, milline näeb välja navigeerimisriba väljalogituna ja sisselogituna.

Joonis 11. Navigeerimisriba väljalogituna



Joonis 12. Navigeerimisriba sisselogituna

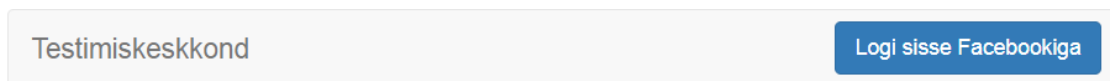


Järgnevalt kirjeldatakse igal lehel toimuvat ja lisatakse visuaalseid näidiseid.

5.3.1 login.html

Navigeerides lõputöös loodud keskkonda, satub kasutaja *login.html* lehele. Kasutajale kuvatakse järgnev vaade:

Joonis 13. *login.html* vaade



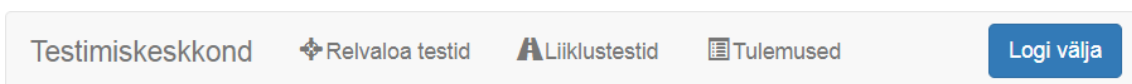
Testimiseks palun logige sisse!

Selleks, et kasutajal oleks võimalik testimiseks asuda, tuleb tal sisse logida kasutades lehe päises olevat nuppu „Logi sisse Facebookiga“. Avanev vaade on kirjeldatud 5.3.2 peatükis.

5.3.2 home.html

Sisselogituna tekib kasutajal võimalus navigeerida lisandunud relvaloa testide, liiklustestide ja tulemuste sektsiooni. Nüüd on navigeerimisribal “Logi sisse Facebookiga” asemel “Logi välja”.

Joonis 14. *Home.html* vaade



Saame targaks

Vajutades navigeerimisribal “Liiklustestid” peale, suunatakse kasutaja edasi “/trafficTests” lõpp-punkti. Selle võtab vastu meetod *WebController.java* failis GET päringuga. Nüüd koostatakse vaade, mis määratakse tagastama *authenticated* kaustast *trafficTests.html* lehte.

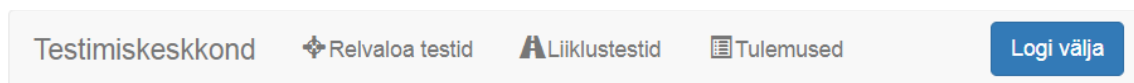
Joonis 15. *trafficTests.html* tagastamine

```
@RequestMapping(value = "/trafficTests", method = RequestMethod.GET)
public ModelAndView trafficTestsView() {
    ModelAndView view = new ModelAndView();
    view.setViewName("authenticated/trafficTests");
    return view;
}
```

Tekkinud vaade on kirjeldatud peatükis 5.3.3.

5.3.3 traffictests.html

Joonis 16. *traffictests.html* vaade



Õpid liiklema?

Alusta testimist

Edasiarenduse huvides on vaatele rakendatud liiklustestide lahendamiseks mõeldud startnupu valmidus.

Vajutades lehe päises “Relvaloa testid” peale, avaneb peatükis 5.3.4 kirjeldatud vaade.

5.3.4 firearmsTesting.html

Relvaloa testidele vastajale kuvatakse alustuseks nupp “Alusta testimist”. Sellele nupule vajutades koostatakse algoritmita rakenduses kasutajale kümnest suvalisest küsimusest koosnev küsimustik, milles igal küsimusel on neli valikvastust. Genereeritud kümnest suvalisest küsimusest tehakse uus *TestFormHelper* objekt, mille küljest hiljem Thymeleaf abil *firearmsTesting.html* lehel küsimused koos vastustega välja näidatakse.

Joonis 17. *firearmsTesting.html* vaade. Relvaloa testi näidise mõttes on kuvatud üks küsimus.

Testimiskeskond Relvaloa testid Liiklustestid Tulemused Logi välja

Õpid relvaeksamiks?

1. Tsiivilkäibes lubatud laskemoon on

- tulirelva padrun, mille kuulil on soomustlähbistav kõvasüdamik
- õõnsaotsalise kuuliga püstoli- ja revolvipadrun
- kesktulepadrun
- ääretulepadrun

Lõpeta

Peale küsimustikule vastamist peab respondent tulemuste salvestamiseks ja kuvamiseks vajutama „Lõpeta“. Küsimustik saadetakse POST meetodiga *WebController.java* faili „/submitFirearmsTest“ lõpp-punkti. Siin kontrollitakse edastatud kasutaja vastusevalikud üle ning salvestatakse lõpuks *SubmittedTest* objektiks. Selle lõpp-punkti lõpuks määratakse uueks HTML leheks *authenticated* kaustast *results.html*. Kaasa antakse eelnevalt salvestatud *SubmittedTesti* objekt *submittedTest*.

Tekkinud vaates (Joonis 18) näeb respondent õigeid ja valesid valikuid, punkte ning kellaega, millal küsimustikule vastati.

Joonis 18. *results.html* esitatud vastuste vaade

Testimiskeskond Relvaloa testid Liiklustestid Tulemused Logi välja

Testi esitamise aeg: 22/05/2018 03:34:43

0 / 10

1. Tsiivilkäibes lubatud laskemoon on

- tulirelva padrun, mille kuulil on soomustlähbistav kõvasüdamik ✘
- õõnsaotsalise kuuliga püstoli- ja revolvipadrun ✘
- kesktulepadrun ✘
- ääretulepadrun ✔

5.3.5 results.html

Vajutades lehe päises “Tulemused” peale, navigeeritakse *results.html* lehele. Siin kuvatakse kasutajale kõik tema vastatud küsimustikud koos kuupäeva ja kellaajaga, millal küsimustikule vastati, testi tüüp ja õigete vastuste arvu kümnest küsimusest.

Joonis 19. *results.html* kõikide tulemuste vaade

Testimiskeskond	Relvaloa testid	Liiklustestid	Tulemused	Logi välja
Esitamise aeg	Testi nimi	Tulemus		
22/05/2018 03:34:43	RELVAEKSAM	0 / 10	Näita tulemusi	
22/05/2018 00:22:01	RELVAEKSAM	0 / 10	Näita tulemusi	
21/05/2018 20:46:48	RELVAEKSAM	0 / 10	Näita tulemusi	
21/05/2018 20:38:37	RELVAEKSAM	1 / 10	Näita tulemusi	

“Näita tulemusi“ viib edasi selle konkreetse küsimustiku tulemuseni ehk siis kuvatakse küsimustikus esinenud küsimused koos valitud vastustega.

6 Valideerimine

Käesolevas lõputöös rakendusele püstitatud nõuded said kõik täidetud.

Kuna lõputöö raames loodud rakendus oli aktiivses arenduses, siis koostas autor valideerimise eesmärgil www.surveymonkey.com keskkonnas rakendusevälise testimise. Testi läbiviimiseks koostati kuus testi viie samasuguse küsimusega.

Esimesed kolm testi olid mõeldud respondentidele, kes vastasid esimesel päeval samale testile kaks korda järjest. Ülejärgmisel päeval lahendati kolmas test. Tulemused on välja toodud allolevas tabelis.

Tabel 2. Tuupimise taktikaga saadud tulemused

Tuupimine	Test 1 (Päev 1)	Test 2 (Päev 1)	Test 3 (Päev 3)
Respondent 1	70%	100%	100%
Respondent 2	70%	90%	80%
Respondent 3	40%	70%	70%
Respondent 4	60%	70%	80%
Keskmine protsent	60%	83%	83%

Viimased kolm testi olid ajastatud õppivatele respondentidele, kes kolme päeva jooksul lahendasid iga päev ühe testi. Tulemused on välja toodud allolevas tabelis.

Tabel 3. Ajastatud õppimise taktikaga saadud tulemused

Ajastatud õppimine	Test 1 (Päev 1)	Test 2 (Päev 2)	Test 3 (Päev 3)
Respondent 5	40%	90%	80%
Respondent 6	40%	90%	100%
Respondent 7	40%	20%	50%

Respondent 8	40%	40%	100%
Keskmine protsent	40%	60%	83%

Tulemuste põhjal võib väita, et kolme päevaga ja kaheksa respondendiga ei kujunenud välja ühe versiooni kindel eelis teise ees ega saavutatud oodatud tulemust. Põhjuseks võib olla testi vastuste läbi vaatamata jätmise, vähene huvi teema vastu, ajanappus. Läbiviidud testid leiab Lisas 1.

7 Võimalikud arengusuunad

Loodud rakendusele on võimalik funktsionaalsust juurde arendada. Järgnevalt kirjeldatakse mõned ideed.

7.1 Küsimuste hulga muutmine

Rakendusele on võimalik juurde ehitada süsteem, mille abil saab kasutaja valida kui mitu küsimust testis kuvada. Selline funktsionaalsus vähendab väheste küsimuste puhul ühes testis infomassi ning võib parandada meeldejätmist. Kas realselt ka nii on, saab tuvastada kogutud tulemuste põhjal.

7.2 Sisselogimise valikuvariandid

Hetkel on rakenduses OAuth baasil realiseeritud ainult Facebook kaudu registreerimine ja sisselogimine. Kasutajate mugavuse mõttes on hea, kui olemas oleks ka Google kontoga registreerimine ja sisselogimine. Kumbkit kontot mitte omavate või nende jaoks, kes ei soovi Facebooki või Google kaudu siseneda, peaks olema ka registreerimisvorm. Lõputöös realiseeriti ainult üks registreerimis- ja sisselogimise meetod ja seda sellepärast, et kasutaja ei teeks andmete kogumise ajal mitut kontot, mis hajutab kasutaja tulemusi analüüsi suhtes.

7.3 Muud testid

Lõputöös kasutati ajastatud õppimise algoritmi katsetamiseks relvaloa harjutusküsimustest koostatud teste. Laienemise mõistes oleks võimalik juurde ehitada näiteks liiklustestide sektsioon või muud seadustest tuletatavad testid.

7.4 Efektivsem algoritm

Tulemustemahu suurenedes selgineb ka lõputöös realiseeritud algoritmi täpsem efektiivsus. Vähesel efektiivsuse puhul on võimalik parema õpitulemuse huvides rakendada enam tagasisidet ja heakskiitu leidnud algoritmi, näiteks SuperMemo2.

8 Kokkuvõte

Töö eesmärgiks oli luua kaks sarnast testimiskeskonda ning võrrelda kahe keskkonna testimistulemusi. Ühele nendest oli rakendatud ajastatud õppimise algoritm. Veebirakenduse eesmärgid said täidetud. Sõltumata eesmärkide täitmisest, pole rakenduse andmebaasi kaudu võimalik respondentide tulemusi võrrelda, sest testimiskeskonna andmebaas on realiseeritud lokaalselt. Probleemi lahendamiseks lõi autor näidisküsimustega testid www.surveymonkey.com keskkonnas ja võrdles seal saadud tulemusi.

Peamiseks probleemseks kohaks oli eestikeelse lõputöö tarbeks erinevate ingliskeelsete terminite vastete leidmine.

Edasiarendamise võimalusega rakendus loodi Javas, kasutades Spring Booti. Keskkonna täiustamise mõttes oleks kasulik esmalt lisada erinevaid registreerimis- ja sisselogimisvõimalusi. Samuti tekiks palju kasutajaid juurde ka liiklustestide realiseerimisel.

Mitmed kasutatud tehnoloogiad olid autori jaoks uued, kuid tulid kasuks kogemuse saamisel.

Loodud rakenduste lähtekoodid on saadaval repositooriumist (vt. Lisa 2).

Kasutatud kirjandus

- [1] University of Waterloo, „Curve of forgetting,“ [Võrgumaterjal]. Available: <https://uwaterloo.ca/campus-wellness/curve-forgetting>. [Kasutatud 20 mai 2018].
- [2] Wikipedia, „Wikipedia – Spaced learning,“ 10 september 2016. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Spaced_learning. [Kasutatud 9 mai 2018].
- [3] Pivotal Software, Inc., „Spring Framework,“ [Võrgumaterjal]. Available: <https://projects.spring.io/spring-framework/>. [Kasutatud 9 mai 2018].
- [4] Spring Initializr. [Võrgumaterjal]. Available: <https://start.spring.io/>. [Kasutatud 9 mai 2018].
- [5] solid IT gmbh, „DB-Engines Ranking,“ [Võrgumaterjal]. Available: <https://db-engines.com/en/ranking>. [Kasutatud 10 mai 2018].
- [6] Pivotal Software, Inc., „Spring Security,“ [Võrgumaterjal]. Available: <https://projects.spring.io/spring-security/>. [Kasutatud 9 mai 2018].
- [7] “Spring Boot Reference Guide,” 2013-2014. [Võrgumaterjal]. Available: <https://docs.spring.io/spring-boot/docs/1.2.1.RELEASE/reference/htmlsingle/#using-boot-starter-poms>. [Kasutatud 9 mai 2018].
- [8] R. Borowiec, „Thymeleaf Page Layouts,“ [Võrgumaterjal]. Available: <https://www.thymeleaf.org/doc/articles/layouts.html>. [Kasutatud 10 mai 2018].
- [9] Riigi Infosüsteemi Amet, „Veebirakendused,“ [Võrgumaterjal]. Available: <https://www.ria.ee/public/ISKE/Veebirakendused.pdf>. [Kasutatud 9 mai 2018].
- [10] D. Elmes, “Anki – documentation,” [Võrgumaterjal]. Available: <https://apps.ankiweb.net/docs/manual.html#new-cards>. [Kasutatud 11 mai 2018].
- [11] D. Elmes, “Anki – documentation,” [Võrgumaterjal]. Available: <https://apps.ankiweb.net/docs/manual.html#reviews>. [Kasutatud 11 mai 2018].

- [12] Anki. [Võrgumaterjal]. Available: <https://apps.ankiweb.net/>. [Kasutatud 11 mai 2018].
- [13] Wikipedia, „Wikipedia – Flash animation,“ 3 mai 2018. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Flash_animation. [Kasutatud 11 mai 2018].
- [14] P. Wozniak, „Application of a computer to improve the results obtained in working with the SuperMemo method,“ 10 mai 1998. [Võrgumaterjal]. Available: <https://www.supermemo.com/english/ol/sm2.htm>. [Kasutatud 12 mai 2018].
- [15] Holger Brandl, „OpenCards – help,“ [Võrgumaterjal]. Available: <http://opencards.info/help.html>. [Kasutatud 13 mai 2018].
- [16] Wikipedia, „Wikipedia – Leitner system,“ 10 märts 2018. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Leitner_system. [Kasutatud 17 mai 2018].
- [17] The Mnemosyne Project, “Principles,” [Võrgumaterjal]. Available: <https://mnemosyne-proj.org/principles.php>. [Kasutatud 13 mai 2018].
- [18] SuperMemo.com, [Võrgumaterjal]. Available: https://www.supermemo.com/en#section_old_smcom. [Kasutatud 13 mai 2018].
- [19] The SuperMemo Team, “The “Repeat” section and 3 buttons for assessment,” 22 juuni 2016. [Võrgumaterjal]. Available: <https://www.supermemo.com/en/blog/repeat-section-and-3-buttons-assessment>. [Kasutatud 12 mai 2018].
- [20] Fresh Memory, „fresh-memory.com – documentation,“ [Võrgumaterjal]. Available: <http://fresh-memory.com/docs/introduction.html#creating-cards>. [Kasutatud 13 mai 2018].
- [21] National League of Cities, „The Athenian Oath,“ [Võrgumaterjal]. Available: <https://www.nlc.org/the-athenian-oath>. [Kasutatud 16 mai 2018].
- [22] Woźniak, Gorzelańczyk, Murakowski, „Two components of long-term memory,“ [Võrgumaterjal]. Available: <https://www.ane.pl/pdf/5535.pdf>. [Kasutatud 18 mai 2018].

- [23] Making Duolingo, “How we learn how you learn,” 14 detsember 2016. [Võrgumaterjal]. Available: <http://making.duolingo.com/how-we-learn-how-you-learn>. [Kasutatud 19 mai 2018].
- [24] Tallinna Tehnikaülikool, „Tikumäng,” 3 märts 2015. [Võrgumaterjal]. Available: <https://courses.cs.ttu.ee/pages/ITI0011:Tikum%C3%A4ng>. [Kasutatud 19 mai 2018].
- [25] Wikipedia, “Wikipedia – Forgetting curve,” 8 märts 2018. [Võrgumaterjal]. Available: https://en.wikipedia.org/wiki/Forgetting_curve. [Kasutatud 21 mai 2018].
- [26] K. Klaas, “Mälu ja selle kasutamine keeleõppes,” 2012. [Võrgumaterjal]. Available: http://e-kursus.kiq.ee/dokumendid/KTJylesanne_tulem.pdf. [Kasutatud 17 mai 2018].
- [27] A. E. Kaasik, “Amneesia,” [Võrgumaterjal]. Available: https://www.kliinik.ee/haiguste_abc/amneesia/id-82. [Kasutatud 17 mai 2018].
- [28] Sean H. K. Kang, “Spaced Repetition Promotes Efficient and Effective Learning,” 13 jaanuar 2016. [Võrgumaterjal]. Available: <http://journals.sagepub.com/doi/pdf/10.1177/2372732215624708>. [Kasutatud 17 mai 2018].

Lisa 1

Ajastatud õppimine 1: <https://www.surveymonkey.com/results/SM-CB923GSDDL/>

Ajastatud õppimine 2: <https://www.surveymonkey.com/results/SM-RSXG3GSDDL/>

Ajastatud õppimine 3: <https://www.surveymonkey.com/results/SM-8TP9BGSDL/>

Tuupimine 1: <https://www.surveymonkey.com/results/SM-ZT3V3GSDDL/>

Tuupimine 2: <https://www.surveymonkey.com/results/SM-95QKFGSDL/>

Tuupimine 3: <https://www.surveymonkey.com/results/SM-PHW3GGSDL/>

Lisa 2

Algoritmilise rakenduse lähtekood: <https://bitbucket.org/geirtag/bakatoalgo/src/master/>

Algoritmita rakenduse lähtekood: <https://bitbucket.org/geirtag/bakatoo/src/master/>