

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Kadri Männimets 185205IAIB

**VEEBIRAKENDUS TALTECHI GITLABI  
GRUPIPROJEKTIDEST ÜLEVAATE  
SAAMISEKS JA NENDE VÕRDLEMISEKS**

Bakalaureusetöö

Juhendaja: Ago Luberg  
PhD

Tallinn 2021

## **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: Kadri Männimets

16.05.2021

## **Annotatsioon**

Käesoleva bakalaureusetöö eesmärgiks on luua veebileht Tallinna Tehnikaülikooli GitLabis ([gitlab.cs.ttu.ee](http://gitlab.cs.ttu.ee)) hoitavate projektide ülevaate saamiseks ning võrdlemiseks, samuti võrldemaks osalejate panust grupitööse nendes projektides.

Töö käigus analüüsitakse olemasolevaid sarnaseid veebirakendusi ja kirjeldatakse, kuidas töötab tagarakendus ning millised on olemasolevad rakenduse vaated. Lisaks tuuakse töö lõpus välja mõned ideed, mida võiks tulevikus rakendusele lisada.

Töö tulemusena valmis veebipõhine GitLabi projektide statistika haldamise rakendus, mis võimaldab kasutajatel saada head ülevaadet GitLabi projektide kohapealt.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 31 leheküljel, 8 peatükki, 13 joonist, 0 tabelit.

## **Abstract**

### **Web Application for TalTech GitLabi Group Projects Overview and Comparison**

The purpose of this thesis is to develop a web application for viewing overviews of student projects in TalTech GitLab ([gitlab.cs.ttu.ee](https://gitlab.cs.ttu.ee)). And also for visually comparing the projects and their members' contribution.

In the first half of the thesis, the author analyzes similar existing applications. Then the author describes how the application works on the backend side and how it looks and functions on the frontend side. Finally the author gives a few ideas on what kind of functionalities could be added to the application in future.

For backend technology, Spring Boot and Gradle are used. The chosen frontend framework is Angular.

The thesis is in Estonian and contains 31 pages of text, 8 chapters, 13 figures, 0 tables.

## Lühendite ja mõistete sõnastik

Adobe XD	Tarkvara kasutajaliidese disainimiseks
Angular	TypeScripti raamistik
API	<i>Application Programming Interface</i> , rakendusliides
Avatud lähtekood	<i>Open source</i> , viitab sellele, et tarkvara on mõeldud olema avalikkusele kättesaadav
<i>Commit</i>	(koodi) muudatuste püsivaks salvestamine
CSS	<i>Cascading Style Sheets</i> , veebi disainimise keel
CSS valija	<i>CSS selector</i> , CSS reegli esimene osa, nende abil leitakse HTML elemendid, mida stiilida
CSV	<i>Comma Separated Values</i> , andmevorming
Domeen	<i>Domain</i> , internetis koosneb domeen võrguaadresside rühmast
Esirakendus	<i>Frontend</i> , kasutajale nähtav osa rakendusest
Git	Versioonihaldustarkvara
GitLab	Git-versioonihaldusega IT-projektide majutuse veebiteenus
Gradle	Rakenduse ehitamise automatiseerimise süsteem
Haru	<i>Branch</i> , GitLabis sõltumatu arenduse suund projektis
HTML	<i>Hyper Text Markup Language</i> , veebilehtede loomise keel
<i>Issue</i>	GitLab'i ülesanne
IT	<i>Information technology</i> , infotehnoloogia
Java	Objektorienteeritud programmeerimiskeel
JavaScript	Esirakenduse objektorienteeritud programmeerimiskeel
Koodikatvus	<i>Code coverage</i> , tarkvara testimise mõõt
Ligipääsuluba	<i>Access token</i> , pikk sõne, mida kasutatakse veebiressursile ligi pääsemiseks lubatud isikutele
Lõpp-punkt	<i>Endpoint</i> , liides, mis võimaldab mitmel protsessil omavahel informatsiooni vahetada
MySQL	Andmebaasi haldamise süsteem
Märgitud aeg	GitLabis <i>issue</i> juurde märgitud aeg, mis näitab, kui kaua selle <i>issue</i> 'ga tegeldi

Pakend	<i>Wrapper</i> , andmestruktuur või programm, mis sisaldab teisi andmeid või programme
Pistikprogramm	<i>Plugin</i> , tarkvaramoodul lisafunktsionaalsuse lisamiseks
PNG	<i>Portable Network Graphics</i> , pildivorming
Python	Programmeerimiskeel
Repositoorium	<i>Repository</i> , koodihoidla, salv
REST	<i>Representational State Transfer</i> , tarkvara arhitektuuri stiil
Salv	Repositoorium, koodihoidla
Skript	<i>Script</i> , valmis kirjutatud koodijupp
Spring	Java arendamise raamistik
Spring Boot	<i>Spring</i> raamistiku osa
SVG	<i>Scalable Vector Graphics</i> , pildivorming
Tagarakendus	<i>Backend</i> , esirakendust teenindav rakendus
Teek	<i>Library</i> , valmiskompileeritud alamprogramm
TypeScript	Programmeerimiskeel, mis põhineb <i>JavaScriptil</i>
Tähtpunkt	<i>Milestone</i> , GitLabis viis <i>issue</i> 'de jälgmiseks kindlal ajaperioodil
UI	<i>User Interface</i> , kasutajaliides ehk rakenduse graafiline kavand
URL	<i>Uniform Resource Locator</i> , internetiaadress

# Sisukord

1 Sissejuhatus .....	10
1.1 Motivatsioon ja probleem.....	10
1.2 Töö eesmärk .....	11
1.3 Rakenduse funktsionaalsed nõuded.....	11
1.4 Metoodika.....	13
2 Sarnased valmisolevad rakendused .....	14
2.1 Screenful.....	14
2.2 LinearB .....	15
2.3 GitLab Analytics.....	15
2.3.1 Repository Analytics .....	15
2.3.2 Issue Analytics.....	16
2.4 Analüüs.....	16
3 Rakenduse analüüs .....	17
3.1 Arenduskäik.....	17
3.1.1 Kasutatud GitLab API lõpp-punktid .....	17
3.2 Andmebaas .....	23
4 Rakenduse kirjeldus.....	24
4.1 Konto seadete muutmise vaade .....	24
4.2 Grupi/projekti lisamise vaade.....	25
4.3 Ülevaadete vaated.....	25
4.3.1 Loetelu.....	26
4.3.2 Ülevaade ja võrdlus keskmisega .....	27
4.3.3 Liikmete tabel, filter ja liikme sidumine .....	28
4.3.4 Graafikud ja otsimise lahter.....	29
4.3.5 Kommentaari lahter .....	33
5 Kasutatud tehnoloogiad esirakenduses .....	35
5.1 Angular .....	35

5.2 Apexcharts .....	35
5.3 Angular Material.....	36
6 Kasutatud tehnoloogiad tagarakenduses.....	37
6.1 Spring Boot.....	37
6.2 Gradle .....	37
6.3 MySQL andmebaas .....	37
7 Rakenduse võimalikud edasiarendused .....	39
7.1 Kõik GitLabi domeenid .....	39
7.2 Sisselogimine GitLabi kaudu.....	39
7.3 Jagamine .....	40
8 Kokkuvõte .....	41
Kasutatud kirjandus .....	42
Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks .....	44
Lisa 2 – Andmebaasiskeem .....	45
Lisa 3 – GitLab API projekti päringu vastuse näide .....	46
Lisa 4 – GitLab API projekti <i>issue</i> märkmete päringu ühe märkme vastuse näide.....	49



## Jooniste loetelu

Joonis 1. <i>Commit</i> 'i tegija id saamine läbi tabelite <code>commit</code> , <code>unlisted_name</code> ja <code>member_name</code> .....	21
Joonis 2. GitLab API lõpp-punktide ja andmebaasi tabelite seos. ....	23
Joonis 3. Konto seadete vaade.....	24
Joonis 4. Grupi/projekti lisamise vaade.....	25
Joonis 5. Grupi loetelu vaade. ....	27
Joonis 6. Projekti ülevaade ja võrdlus keskmisega vaade. ....	28
Joonis 7. Liikmete table ja liikme sidumise vaade. ....	29
Joonis 8. Graafiku detailvaade.....	30
Joonis 9. Neljaosaline graafikute komponent.....	31
Joonis 10. Graafiku vaade koos allalaadimisvõimalustega. ....	32
Joonis 11. Graafikute komponent koos otsimissektsiooniga.....	32
Joonis 12. Otsimissektsioon tähtpunktide järgi otsimise võimalusega. ....	33
Joonis 13. Projekti/Liikme kommentaari vaade. ....	34

# 1 Sissejuhatus

GitLab<sup>1</sup> on üks populaarsemaid arendajate tööriistu, millel on hinnanguliselt üle 30 miljoni registreeritud kasutaja [1]. GitLabil on palju alternatiivne, neist populaarseimad GitHub<sup>2</sup>, GitBucket<sup>3</sup> ning SourceForge<sup>4</sup> [2]. Antud töös luuakse veebirakendus just Tallinna Tehnikaülikooli GitLabi projektide ülevaateks ja võrdlemiseks. Kuigi rakendus on loodud TalTechi GitLabiga töötama, saab seda kergeste muudatustega panna töötama ka kõikide GitLabi domeenidega, sellest on täpsemalt juttu peatükis 7.

Selles peatükis antakse ülevaade motivatsioonist ja probleemist, töö eesmärgist, rakenduse funktsionaalsetest nõutest ning töö metoodikast.

## 1.1 Motivatsioon ja probleem

Ülikoolis ja mujalgi kasutatakse GitLabi koodipõhiste projektide haldamiseks, seda nii individuaalprojektide kui ka grupiprojektide puhul. Individuaalprojektide puhul on osaliste panus selge, sest projekti teeb ainult üks inimene, kuid puudub võimalus võrrelda osaliste panust grupiprojektide puhul ning projekte omavahel.

Teema on juhendaja poolt pakutud, sest juhendaja tundis puudust niisuguse rakenduse vastu, et paremini hallata enda poolt läbi viidavat ainet, kus tudengid teevad tööd grupis ning nende panuseid võib olla kohati keeruline võrrelda.

Siiani on kasutatud selliste võrdluste ja ülevaadete saamiseks Pythoni skripti, mis ei ole avatud lähtekoodiga. Skripti kasutamine on ebamugav, sest see tuleb ise oma arvutis tööle

---

<sup>1</sup> <https://about.gitlab.com/>

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

<sup>3</sup> <https://bitbucket.org/>

<sup>4</sup> <https://sourceforge.net/>

panna, mis nõuab teatavat tarkvara ning lisaks on sealt andmete lugemine keeruline ja ebamugav, sest andmed kuvatakse konsooli lihttekstina.

## **1.2 Töö eesmärk**

Käesoleva lõputöö eesmärgiks on luua veebirakendus, mis kuvab kasutajale ülevaadet kasutaja poolt soovitud GitLabi projektidest. Eesmärk on luua rakendus eelkõige just grupitööde ja nendes osalenute panuse paremaks ülevaateks ning võrdlemiseks.

Rakenduses saab kasutaja jaotada projektid gruppidesse, mitmesalveliste projektide puhul salved alamgruppidesse.

## **1.3 Rakenduse funktsionaalsed nõuded**

Selles peatükis toob autor välja toimingud, mida kasutaja peab saama veebiteenuse kaudu teha.

- Registreerub, sisestades kasutajatunnuse ja parooli kaks korda.
- Logib sisse, sisestades kasutajatunnuse ja salasõna.
- Logib välja.
- Uuendab oma kontoga seotud GitLabi ligipääsuluba, sisestades GitLabi ligipääsuloa
- Muudab parooli, sisestades praeguse parooli ja uue parooli kaks korda
- Lisab uue grupi, sisestades grupi nime.
- Lisab uue alamgrupi, valides ülemgrupi ja sisestades loodava alamgrupi nime.
- Lisab uue(d) projekti(d) gruppi(desse), valides grupi(d) ja projekti(d).
- Eemaldab projekti grupist.
- Kustutab grupi.

- Näeb projekti ülevaadet ja võrdlust keskmise projektiga selles grupis, kus on toodud välja *issue*'de arv, liikmete poolt märgitud aeg ja *commit*'ide arv.
- Näeb projekti liikmete ülevaadet, kus on toodud välja liikme märgitud aeg ning *commit*'ide arv.
  - Saab antud ülevaadet filtreerida aja järgi, valides alguskuupäeva ja lõppkuupäeva.
  - Saab antud ülevaadet filtreerida projekti tähtpunktide järgi, valides vähemalt ühe tähtpunkti.
- Näeb graafikut projekte *commit*'ide kohta, kus on liikmed eraldi välja toodud.
- Näeb graafikut projekti liikmete poolt märgitud aja kohta, kus on liikmed eraldi välja toodud.
- Saab kuvada lisagraafikuid projekti *commit*'ide ja liikmete poolt märgitud aja kohta, kus on liikmed eraldi välja toodud, valides alguskuupäeva ja lõppkuupäeva või vähemalt ühe tähtpunkti.
- Saab kirjutada projekti kohta kommentaari.
- Saab projekti kohta kirjutatud kommentaari muuta.
- Näeb liikme ülevaadet ja võrdlust keskmise liikmega selles projektis, kus on välja toodud liikme poolt märgitud aeg ja *commit*'ide arv.
- Näeb graafikut liikme *commit*'ide kohta.
- Näeb graafikut liikme poolt märgitud aja kohta.
- Saab kuvada lisagraafikuid liikme poolt märgitud aja ja *commit*'ide kohta, valides alguskuupäeva ja lõppkuupäeva.
- Saab kirjutada liikme kohta kommentaari.
- Saab liikme kohta kirjutatud kommentaari muuta.

## 1.4 Metoodika

Töö käigus tehakse eesmärgi saavutamiseks esialgu taustauuring olemasolevate sarnaste veebirakenduste osas. Seejärel tehakse valmis esialgne veebilehe prototüüp, mille järgi hakatakse ka kohe esirakendust realiseerima.

Esirakenduse realiseerimiseks on kasutatud Angular<sup>1</sup> raamistikku. Angular on avatud lähtekoodi esirakenduse arendamise raamistik, mis kasutab TypeScripti<sup>2</sup>. Autor valis esirakenduse tehnoloogiaks Angulari, sest tal on sellega varasemaid kogemusi.

Tagarakendus on üles ehitatud Spring Boot<sup>3</sup> raamistikul, kasutades Java<sup>4</sup> programmeerimiskeelt. Spring Boot on avatud lähtekoodiga tagarakenduse arendamise raamistik. Selle raamistiku valik oli samuti tingitud eelnevast kogemusest.

Andmebaasi tehnoloogiaks on valitud MySQL<sup>5</sup>, mis on andmebaasi haldamise süsteem, samuti avatud lähtekoodiga. Andmebaasi valikul olid sõelal MySQL ja PostgreSQL<sup>6</sup>, neist valituks osutus MySQL, sest see oli autorile juba varasemast tuttav.

Oskustermine eesti keelde tõlkimisel kasutas autor Cybernetica standardipõhist tarkvaratehnika sõnastikku<sup>7</sup> ning Heikki Vallaste ingliskeelsete info- ja sidetehnoloogia terminite seletavat sõnaraamatut<sup>8</sup>.

---

<sup>1</sup> <https://angular.io/>

<sup>2</sup> <https://www.typescriptlang.org/>

<sup>3</sup> <https://spring.io/projects/spring-boot>

<sup>4</sup> <https://www.oracle.com/java/>

<sup>5</sup> <https://www.mysql.com/>

<sup>6</sup> <https://www.postgresql.org/>

<sup>7</sup> <https://stats.cyber.ee/>

<sup>8</sup> <http://www.vallaste.ee/index.asp>

## 2 Sarnased valmisolevad rakendused

GitLabi projektide statistika haldamiseks ja analüüsimiseks on tehtud mitu erinevat rakendust. Antud peatükis kirjeldab autor neist kolme.

### 2.1 Screenful

Screenful<sup>1</sup> on veebirakendus, mis võimaldab luua raporte ja graafikuid erinevate koodihaldamiskeskondade, k.a GitLabi, gruppide ja projektide kohta. Lisaks annab Screenful väga hea ja põhjaliku ülevaate projekti *issue*'de kohta. Screenful sobib pigem arendajatele oma progressi jälgimiseks.

Rakenduse peamiseks ressursiks ülevaadete ja kokkuvõtete kuvamiseks on *issue*'d, mis tähendab, et pea kõik graafikud ja raportid koosnevad *issue*'dest - sellest, millal nad on avatud, kui kaua on nagu mingis olekus olnud, millised sildid on neil küljes on, kes nendega tegelenud on ja millal kinni on pandud.

Screenfuli kasutamine on tasuta esimesed 21 päeva, pärast seda saab valida mitme erineva tasulise programmi vahel, neist odavaim 39 \$ kuus ning kallim 799 \$ kuus. Odavamte puhul saab graafikuid, raporte ja andmete ajalugu piiratud arvul, kõige kallima puhul saab graafikuid ja raporte teha lõpmata palju, andmed arvestatakse kokku kogu aja peale, mitte lihtsalt viimase paari kuu, ning lisaks on olemas Premium abi.

Screenfuli positiivseks pooleks on väga suur hulk erinevaid graafikuid ja raporte, puudujäägiks väga kõrge hind ning puudub võimalus võrrelda omavahel nii projekte kui ka liikmete panust märgitud aja ja *commit*'ide kohapealt.

---

<sup>1</sup> <https://screenful.com/>

## 2.2 LinearB

LinearB<sup>1</sup> on erinevate veebipõhiste salvede, sealhulgas GitLabi, andmete analüüsimiseks tehtud veebileht, mis on orienteeritud rohkem arendajate tiimile.

LinearB miinusteks on võrdlemisi pikk ja keeruline tasulisele versioonile ülendamise protsess ning muidugi ei ole see ka päris tasuta. Tasuta programm hõlmab endas 45 päeva vältel andmeid, 50 salve, maksimaalselt üht tiimi ning kaheksat arendajat. Tasulise programmi puhul, mis maksab 45 \$ arendaja kohta kuus, võib arendajaid, tiime ja salvesid olla piiramatus koguses, kuid andmeid saab ainult kolme aasta kohta. LinearB ei anna ülevaadet tiimi liikmete panustest individuaalselt ehk seal ei saa jälgida märgitud aega ega *commit*'e ühe tiimi liikme kohta.

## 2.3 GitLab Analytics

GitLab Analytics<sup>2</sup> on GitLabi sisseehitatud analüüsimissüsteem, millel on palju erinevaid komponente, millest valdav osa nõuab tasulise GitLabi konto olemasolu (nt Premium). Selles peatükis analüüsib autor neist kahte komponenti, mis sarnanevad funktsionaalsuselt antud lõputööle.

GitLab Analyticsi suur pluss on see, et see on olemas iga GitLabi projekti küljes ilma, et selle lisamiseks peaks ekstra samme tegema (konto uude kohta).

### 2.3.1 Repository Analytics

GitLabi Repository Analytics<sup>3</sup> on saadaval GitLabi tasuta versioonis ning annab ülevaate kolmest asjast:

- projektis kasutatud programmeerimiskeeltest protsentuaalselt (baitide järgi mõõdetud, välja on arvatud genereeritud ja mujalt hangitud kood).

---

<sup>1</sup> <https://linearb.io/>

<sup>2</sup> <https://docs.gitlab.com/ee/user/analytics/>

<sup>3</sup> [https://docs.gitlab.com/ee/user/analytics/repository\\_analytics.html](https://docs.gitlab.com/ee/user/analytics/repository_analytics.html)

- koodikatus (ingl *code coverage*), annab graafiku kujul ülevaate testide koodi katvusest kuupäeva kohta, kus madalaim väärtus on 0% ja kõrgeim 100%.
- *commit*'ide statistika haru (ingl *branch*) kohta (arvestab maksimaalselt 2000 *commit*'i): kui palju on *commit*'e kokku, mitu *commit*'i on päevas keskmiselt tehtud ning mitme autori poolt. Lisaks kuulub selle ülevaate alla kolm graafikut, mis kajastavad pigem üldist *commit*'imise statistikat – *commit*'ide arv päeva kohta kuus, *commit*'ide arv nädalapäeva kohta ning *commit*'ide arv tunni kohta ööpäevas. Kõik need graafikud kajastavad *commit*'ide arvu kõikide osalejate kohta kokku, mitte eraldi.

### 2.3.2 Issue Analytics

GitLabi Issue Analytics<sup>1</sup> on saadaval GitLabi tasulises versioonis (Premium) ning võimaldab vaadata kahte peamist asja:

- graafikut, mis näitab, mitu *issue*'t on mingis kuus avatud. Andmeid on võimalik filtreerida autori, määratu, tähtpunkti, sildi, kasutaja reaktsiooni ning kaalu järgi.
- tabelit, mis näitab 100 *issue*'t, kus iga *issue* kohta on tabelis veerud *issue* vanuse, staatuse, tähtpunkti, kaalu, tähtaja, määratu ning *issue* avaja kohta.

Puudub ülevaade *issue* jaoks kulutatud ajast.

## 2.4 Analüüs

Ükski kolmest eelnevalt kirjeldatud veebirakendusest ei ole mõeldud projektide omavaheliseks võrdlemiseks ning projekti liikmete panuste võrdlemiseks. Kõik kolm rakendust on mõeldud pigem arendajate tiimile oma progressi jälgimiseks, kuid mitte kolmandale osapoolle nende võrdlemiseks. Rakenduste teiseks ühiseks negatiivseks jooneks on see, et nad on tasulised.

---

<sup>1</sup> [https://docs.gitlab.com/ee/user/analytics/issue\\_analytics.html](https://docs.gitlab.com/ee/user/analytics/issue_analytics.html)



## 3 Rakenduse analüüs

Selles peatükis annab autor ülevaate arenduskäigust ja andmebaasist ning seletab, kuidas rakendus töötab.

### 3.1 Arenduskäik

Rakenduse arendus algas autoripoolse analüüsimisega rakenduse nõuetele ning kasutatavate tehnoloogiate valimisega. Kuna autoril on varasem kogemus Angulari ja Spring Booti tehnoloogiatega, siis valis ta need ka selle projekti arendamiseks.

Enne rakenduse arendamist tegi autor valmis rakenduse esialgse prototüübi, kasutades selleks loodud tarkvara Adobe XD<sup>1</sup>. Rakenduse koodi kirjutamine algas valmis tehtud prototüübi kasutajaliidese realiseerimisega esirakenduses.

Kui esialgne kasutajaliides (UI, ingl *user interface*) oli valmis, oli ka selge, missuguseid andmeid ja mis kujul on vaja tagarakenduselt küsida. Selle informatsiooni põhjal tegi autor tagarakenduse projekti ning esialgsed andmed ei liikunud mitte läbi andmebaasi, vaid otse läbi GitLab API<sup>2</sup>, seda selleks, et GitLab API-ga praktikas lähemalt tutvuda.

Üsna kiiresti sai selgeks, et andmete otse GitLab API-lt küsimine võtab tohutult palju aega, selle pärast otsustas autor rakendusele vajalikud andmed salvestada andmebaasi.

#### 3.1.1 Kasutatud GitLab API lõpp-punktid

Selles rakenduses kasutas autor kõikide andmete kogumise jaoks GitLab API lõpp-punkte. Lõpp-punkt (ingl *endpoint*) on sidekanali üks ots, teisisõnu asukoht, kust rakendusliides saab enda funktsiooni täitmiseks vajalikke andmeid [3].

Selleks, et GitLab API-lt üldse andmeid saada, tuleb iga päringuga anda kaasa parameeter `access_token`, milleks on GitLabi kasutajaga seotud juurdepääsuluba, mille järgi teab

---

<sup>1</sup> <https://www.adobe.com/products/xd.html>

<sup>2</sup> <https://docs.gitlab.com/ee/api/>

lõpp-punkt, missugustele projektidele antud kasutajal ligipääs on. Selle töö raames kasutas autor järgmisi GitLab API lõpp-punkte:

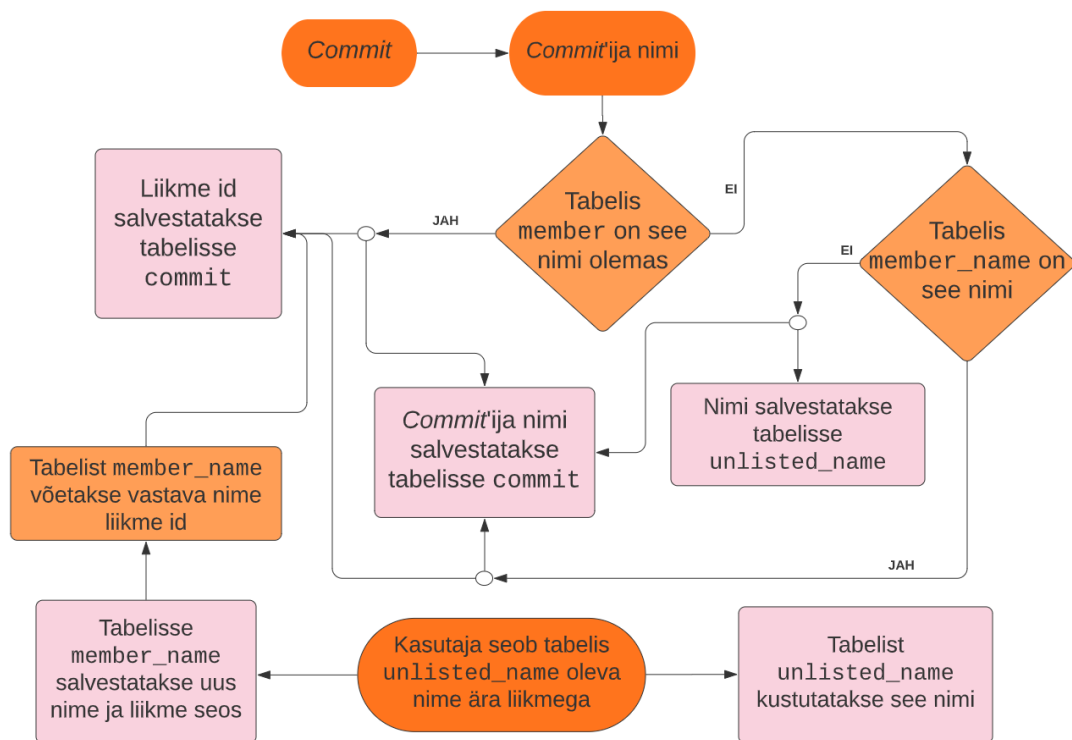
- `/api/v4/projects/` - tagastab kasutaja kõikide projektide loetelu.
- `/api/v4/projects/:projectId` - tagastab projekti id-ga `projectId` peamised parameetrid nagu nimi, kirjeldus, URL ja palju muud. Täpsema ülevaate saamiseks vaadake Lisa 3. Projekti andmed kirjutatakse andmebaasi tabelisse `project`, lõpp-punktid saadavad väärtused, mis sellesse tabelisse kirjutatakse, on projekti id, nimi ning URL.
- `/api/v4/projects/:projectId/members` - projekti id-ga `projectId` kõikide liikmete loetelu. Kõik projekti liikmed salvestatakse andmebaasis tabelisse `member`.
- `/api/v4/projects/:projectId/milestones` - tagastab projekti id-ga `projectId` kõikide tähtpunktide loetelu. Kõik projekti tähtpunktid salvestatakse andmebaasis tabelisse `milestone`. Tähtpunkti puhul salvestatakse lõpp-punktist saadavatest andmetest andmebaasi tema id, nimetus ning algus- ja lõppkuupäev.
- `/api/v4/projects/:projectId/issues` - tagastab projekti id-ga `projectId` kõikide `issue`'de loetelu. Kõik projekti `issue`'d salvestatakse andmebaasis tabelisse `issue`. Salvestatavad parameetrid saadud päringust on `issue` id, olek – kas suletud või avatud – ning tähtpunkti id, kui `issue`'l on tähtpunkt.
- `/api/v4/projects/:projectId/issues/:issueId/notes` - tagastab projekti id-ga `projectId` `issue` id-ga `issueId` kõikide märkmete loetelu. Andmebaasi salvestatakse ainult sellised märkmed, mis on seotud aja märkimisega. Selleks, et tuvastada, kas märke on seotud aja märkimisega, tuleb vaadata märkme keha (ingl *body*). Märkme, mis on seotud aja märkimisega, keha on kujul „added <aeg> of time spent at <kuupäev>“ või „subtracted <aeg> of time spent at <kuupäev>“, kus <> vahel on vastavalt aeg kasutajale mugaval kujul, näiteks „4d 6h 28m“, ning kuupäev

kujul yyyy-mm-dd. Juhul, kui märke on aja märkimise kohta, kirjutatakse selle märkme andmed andmebaasis tabelisse `note`. Märkme puhul salvestatakse andmebaasi tema `id`, tema `issue id`, märkija `id`, märgitud aeg sekundites ning märkme tegemise kuupäev ja kellaaeg. Kuna märgitud aeg ei ole antud sekundites, arvutatakse see sekunditesse ümber. Märgitud ajal on kuus erinevat vormi: mo ehk kuud (ingl *months*), w ehk nädalad (ingl *weeks*), d ehk päevad (ingl *days*), h ehk tunnid (ingl *hours*), m ehk minutid (ingl *minutes*) ning s ehk sekundid (ingl *seconds*). GitLabikohaselt on ühes kuus neli nädalat, ühes nädalas viis päeva, ühes päevas kaheksa tundi, ühes tunnis 60 minutit ning ühes minutis 60 sekundit. Näide lõpp-punktilt saadud ühest märkmest on Lisas 4.

- `/api/v4/projects/:projectId/repository/branches` - tagastab projekti `id`-ga `projectId` kõikide harude loetelu. Projekti harusid ei salvestata andmebaasi, vaid kasutatakse projekti `commit`'ide saamiseks järgmises punktis ära.
- `/api/v4/projects/:projectId/repository/commits` - tagastab projekti `id`-ga `projectId` `commit`'ide loetelu. Kuna projektidel saab olla rohkem kui üks haru (`master`), siis kõikide teiste harude `commit`'ide kättesaamiseks tuleb lõpp-punktile määrata parameetriks `ref_name` vastava haru nimi. Kõik projekti `commit`'id salvestatakse andmebaasis tabelisse `commit`. Andmebaasi salvestatakse:
  - `Commit`'i `id`, tabelisse `commit`
  - `Commit`'imise kuupäev, tabelisse `commit`
  - Liikme `id`, kes selle `commit`'i tegi, tabelisse `commit`. Antud lõpp-punkt ei tagasta `commit`'i teinud liikme `id`. Lõpp-punkt tagastab `commit`'i teinud liikme nime. Nime ja projekti `id` järgi otsitakse tabelist `member` liiget. Juhul, kui liige leitakse, salvestatakse tema `id` mainitud tabelisse. Juhul, kui sellise nimega liiget selle projekti juurest ei leita, vaadatakse tabelisse `member_name`. Tabelis `member_name` on kirjas projekti liikme lisanimi, mille kasutaja on projekti liikmele määranud. Lisanimed tekivad siis, kui projekti on `commit`'i teinud muudetud nimega liige. Sellest tabelist

otsitakse nime, projekti id ja kasutaja id järgi, kas see nimi on mõnele liikmele lisanimeks märgitud. Juhul, kui on, võetakse sealt samast tabelist liikme id ja salvestatakse tabelisse `commit`. Juhul, kui ka tabelist `member_name` ei leita `commit`'i tegija id-d, jääb `commit`'ija id tabelisse `commit` salvestamata.

- `Commit`'ija nimi, tabelisse `commit`. `Commit`'ija nime salvestamine on vajalik seepärast, et antud lõpp-punkt ei tagasta `commit`'ija id-d. Kuna selle `commit`'i autor võib olla muudetud nimega projekti liige ning tabelis `member_name` ei pruugi olla seda nime veel ühelegi liikmele määratud, on vaja `commit`'i tegija nimi ära salvestada. Lisaks salvestatakse `commit`'ija nimi ka tabelisse `unlisted_name`, seda vaid juhul, kui `commit`'i ei autorit ei ole võimalik kohe kindlaks teha. Neid salvestatud nimesid kasutatakse selleks, et kasutaja saaks liikme nime viia kokku liikmega. Kui kasutaja on veebirakenduses liikme nimi liikmega ära sidunud, salvestatakse vastavalt liikme nimi, liikme id, projekti id ja kasutaja id tabelisse `member_name` ning vastav nimi kustutatakse tabelist `unlisted_name`. Nimed, mida kasutajale kokku viimiseks pakutakse, tulevad tabelist `unlisted_name`. Pärast liikme sidumist nimega uuendatakse kõiki tabeli `commit` neid ridu, mille liikme id on puudu ja `commit`'ija nimi on vastseotud nimi. Sellistel ridadel määratakse `commit`'i tegija id-ks vastavalt seotud liikme id (Joonis 1).



Joonis 1. *Commit*'i tegija id saamine läbi tabelite *commit*, *unlisted\_name* ja *member\_name*.

- Juhul, kui *commit* on seotud ühe või rohkema *issue*'ga, salvestatakse tabelisse *commit\_issue* iga *issue* kohta nii *issue* id kui ka *commit*'i id. Seda, kas *commit* on seotud mõne *issue*'ga, saab lugeda *commit*'i sõnumist (ingl *commit message*). *Commit*'i sõnumis saab kirjutada, millise *issue*'ga antud *commit* seotud on. *Commit*'i sõnumisse kirjutatakse viide vastavale *issue*'le kujul #*issueId*, kus *issueId* on mainitud *issue* id. Üks *commit* võib olla seotud nulli või rohkema *issue*'ga. Tabel *commit\_issue* on tabel, mis seob ühe *commit*'i ära kõikide tema *issue*'dega. Seda on vaja seepärast, et *commit* ei ole seotud tähtpunktiga, kuid *issue* on. Niiviisi saab läbi *issue* määrata *commit*'i tähtpunkti. Kui *commit*'i puhul on läbi *issue* teada tema tähtpunkt, saab selle järgi kuvada kasutajale rakenduse vaatesse informatsiooni tähtpunkti *commit*'ide kohta.

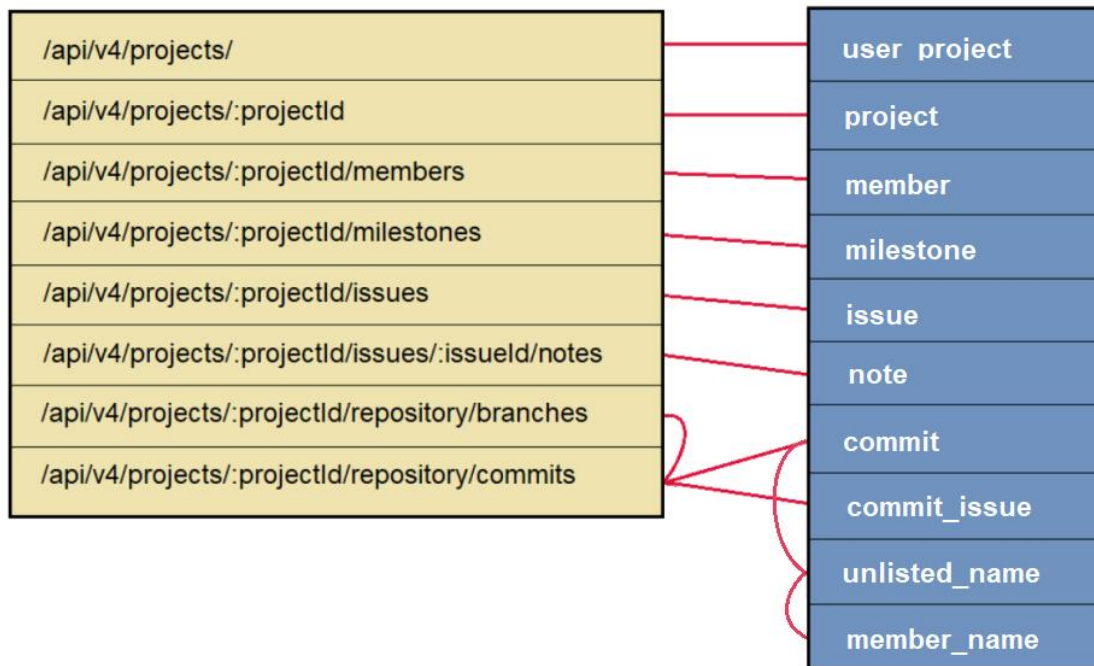
Vaikimisi annab iga GitLab API päring 20 vastust. Selleks, et saada ühe päringuga rohkem kui 20 vastust, tuleb lõpp-punktile määrata parameeter *per\_page* väärtuseks suurem arv. Maksimaalselt tagastavad GitLab API lõpp-punktid korraga 100 vastust. Kui neid vastuseid on aga rohkem anda, siis tuleb võtta kasutusele ka parameeter *page*. Lõpp-punkti parameeter *page* määrab ära, mitmes vastustehulk vastuseks antakse. Kõige

väiksem page väärtus saab olla 1 ning kõige suurem oleneb sellest, kui palju vastuseid on lõpp-punktil anda ning mitu tükki korraga küsitakse ehk parameetri `per_page` väärtusest.

Antud töös on lahendatud kõikide vastuste saamine GitLab API-lt selliselt, et korraga küsitakse alati 100 vastust ehk parameetri `per_page` väärtus on alati 100. Parameeter `page` on alguses 1. Kui lõpp-punktilt saadi 100 vastust, siis suurendatakse parameetrit `page` ja küsitakse uued vastused. Parameetrit `page` suurendatakse ühe võrra senikaua, kuni lõpp-punkt ei tagast enam 100 vastust. Kui lõpp-punkt tagastab alla 100 vastuse, on kõik vastused kätte saadud. Kõikide vastuste GitLab API-lt saamine on lahendatud seesuguselt, sest selliselt tuli vähem päringuid teha. Teine viis kõikide vastuste saamiseks oleks olnud küsida vastuseid seni, kuni lõpp-punkt tagastab null vastust, sest siis on selge, et rohkem pole midagi võtta. Kui selliselt toimida, oleks väga suure tõenäosusega igakord teha üks päring ülearu ning see raisaks mõttetult aega.

Lõpp-punktidest saadud vajalikud andmed on salvestatud andmebaasi (Joonis 2), et neid ei peaks koguaeg GitLab API-lt uuesti küsima. Projekti uuendamisel uuendatakse ka kõiki teisi tabeleid, milles on projektiga seonduvat infot. Ehkki vanad andmed on juba andmebaasis kirjas, tuleb teatud juhtudel siiski ka tabeli kõik projektiga seotud read projekti uuendamisel uuendada:

- tabeli `member` read tuleb uuendada, sest liikme nimi võib muutuda.
- tabeli `milestone` read tuleb uuendada, sest tähtpunktis võib olla muutunud algus- ja/või lõppkuupäev.
- tabeli `issue` read tuleb uuendada, sest `issue` tähtpunkt ja olek võivad muutuda



Joonis 2. GitLab API lõpp-punktide ja andmebaasi tabelite seos.

### 3.2 Andmebaas

Rakenduse andmebaas koosneb 14-st tabelist (vt Lisa 2). Tabel `user` on kasutajate andmete informatsiooni hoidmiseks. Tabelis `user_projects` on kõikide nende projektide id-d ja nimed, millele antud kasutajal on GitLab-is ligipääs. Tabelid `groups` ning `group_projects` hoiavad endas andmeid kasutaja gruppide ja nendesse kuuluvate projektide kohta.

Tabelites `commit`, `issue`, `member`, `milestone` ja `notes` hoitakse informatsiooni projekti vastavate andmete kohta. Tabel `commit_issue` hoiab endas informatsiooni vastavalt `commit`'i ja `issue` sidumise kohta. Tabelis `unlisted_name` hoitakse neid nimesid, mis ei ole veel ühegi liikmega ära seotud, kuid on mingi `commit`'i autoriks. Kui tabelis `commit` mõni veel sidumata `commit`'i autor reaalse kasutajaga projektist ära seotakse, lähevad need andmed tabelisse `member_names`, kus hoitakse kasutajate erinevaid (lisa)nimesid. Tabelis `project` hoitakse informatsiooni projekti ülevaate ja kokkuvõtte kohta.

## 4 Rakenduse kirjeldus

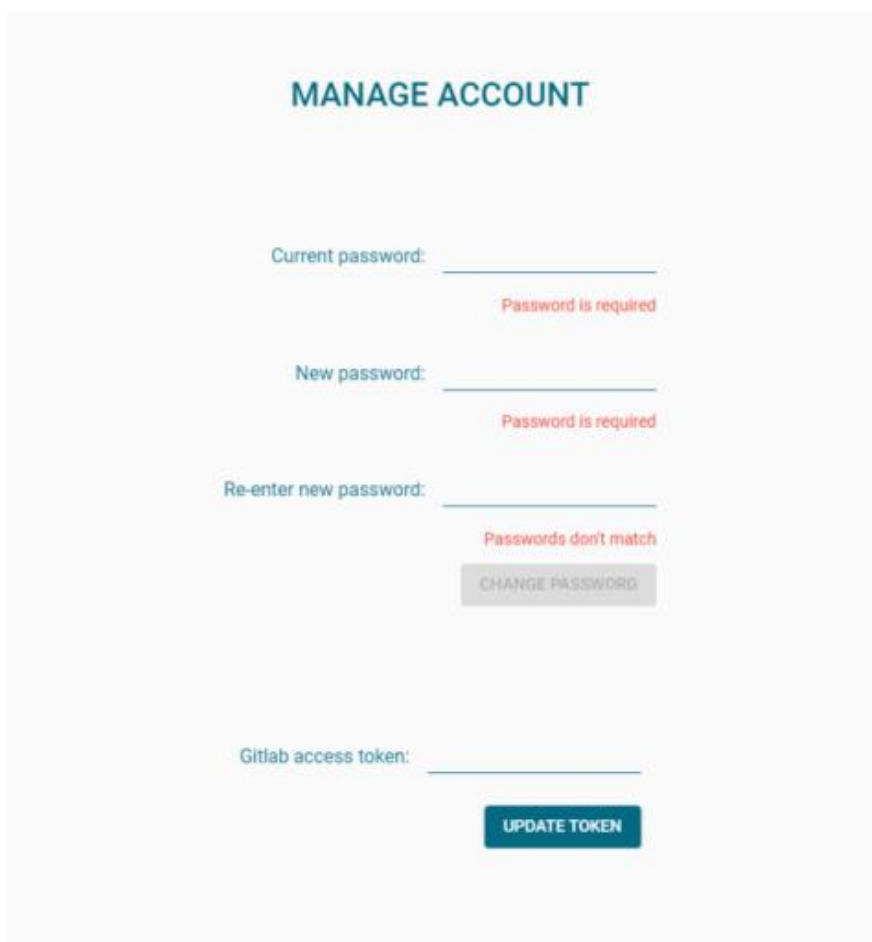
Selles peatükis kirjeldab autor valminud rakenduse vaateid ja funktsionaalsust.

### 4.1 Konto seadete muutmise vaade

Konto seadete muutmise vaate (Joonis 3) alt saab kasutaja muuta oma olemasolevat parooli ning GitLabi juurdepääsuluba (ingl *access token*).

Parooli muutmiseks on vaja sisestada hetkel kehtiv parool ning uus parool kaks korda.

GitLabi juurdepääsuloa muutmiseks on vaja sisestada lahtrisse GitLabi lehelt oma konto seadete alt leitav token.



The screenshot displays a web form titled "MANAGE ACCOUNT". It contains three password input fields and one text input field. The first field is labeled "Current password:" and has a red error message "Password is required" below it. The second field is labeled "New password:" and also has a red error message "Password is required" below it. The third field is labeled "Re-enter new password:" and has a red error message "Passwords don't match" below it. Below these three fields is a grey button labeled "CHANGE PASSWORD". The fourth field is labeled "Gitlab access token:" and has a blue button labeled "UPDATE TOKEN" below it.

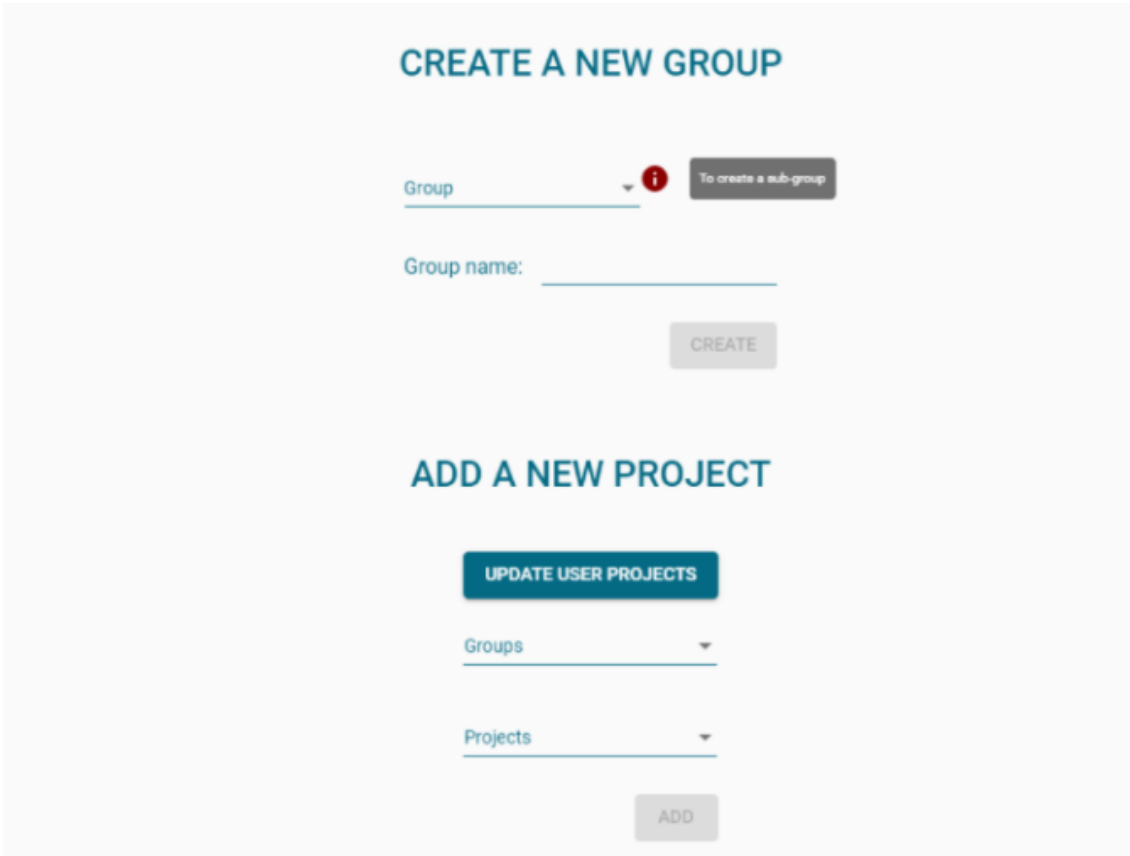
Joonis 3. Konto seadete vaade.



## 4.2 Grupi/projekti lisamise vaade

Et luua uut tavalist gruppi, on tarvis sisestada loodava grupi nimi ja vajutada nuppu “Create”. Kui kasutaja soovib teha uut alamgruppi, tuleb lisaks nime sisestamisele valida gruppide loetelust loodavale alamgruppile nn ülemgrupp. Kasutaja võib luua mitu samanimelist gruppi. Kui grupi loomine õnnestus, kuvatakse kasutaja sellekohane sõnum, kui ei õnnestunud, kuvatakse ka selle kohta sõnum.

Projekti(de) lisamiseks on vaja olemasolevatest gruppidest valida üks või mitu gruppi, kasutaja projektide loetelust projekt(id) ning vajutada nupule “Add”. Projekti saab lisada niisugusesse gruppi, kus seda sama projekti juba olemas ei ole. Kui projekti lisamine õnnestus, kuvatakse kasutajale sellekohane kiri, ebaõnnestumise korral samuti vastav teavitus. Kasutaja projektide loetelu uuendamiseks on nupp “UPDATE USER PROJECTS” (Joonis 4).



The screenshot displays two main sections of the user interface. The top section, titled "CREATE A NEW GROUP", features a dropdown menu labeled "Group" with a red information icon and a tooltip that reads "To create a sub-group". Below this is a text input field for "Group name:" and a "CREATE" button. The bottom section, titled "ADD A NEW PROJECT", includes a prominent "UPDATE USER PROJECTS" button. Underneath are two dropdown menus: "Groups" and "Projects", followed by an "ADD" button.

Joonis 4. Grupi/projekti lisamise vaade.

## 4.3 Ülevaadete vaated

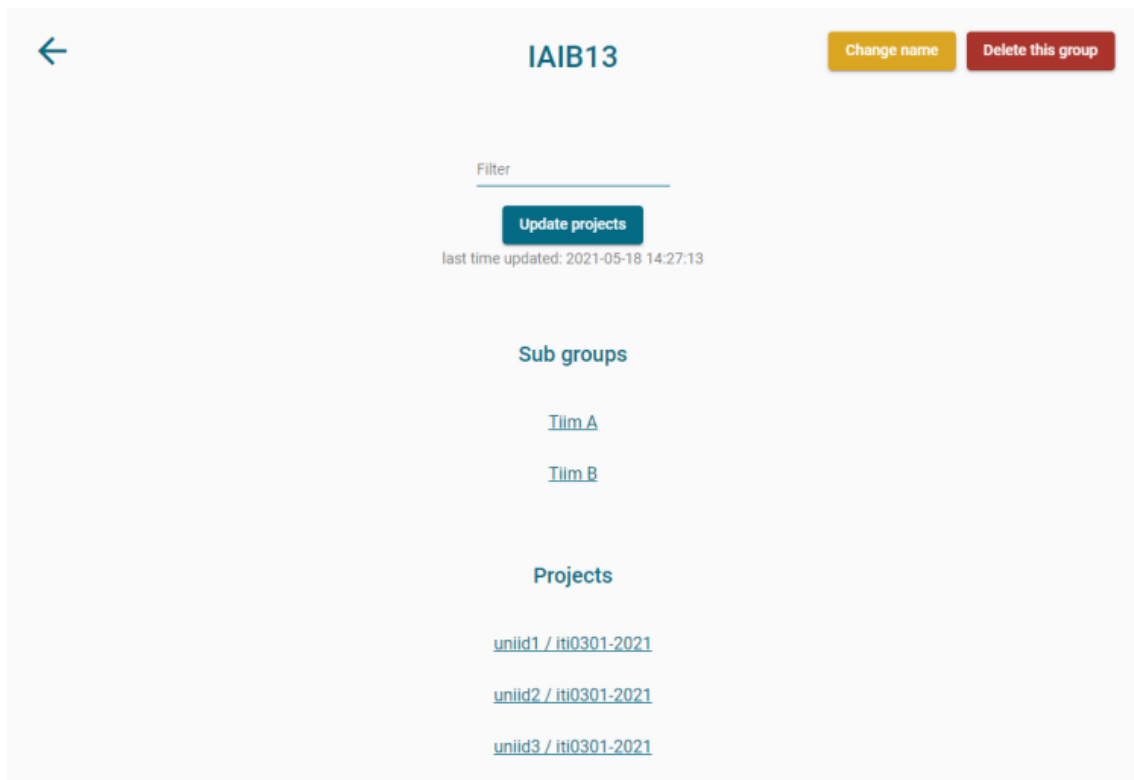
Ülevaadete vaateid on kokku viis:

- kõikide gruppide vaade, mis annab ülevaate igast ülemgrupist ning laseb hõlpsasti võrrelda erinevaid gruppe omavahel
- ühe grupi vaade, mis annab ülevaate grupis olevatest alamgruppidest ja projektist
- alamgrupi vaade, mis annab ülevaate alamgrupis olevatest liikmetest. Alamgrupi vaade on mõeldud eelkõige nendeks juhtudeks, kui üks projekt on jaotatud laiali mitme repositooriumi peale. Tavaliselt on sellistel juhtudel projekti esirakendus ja tagarakendus eraldi ning alamgrupp võtab niimoodi mitme repositooriumi andmed kokku ühte vaatesse
- projekti vaade, mis annab ülevaate projekti liikmetest ja nende panustest grupitöösse
- liikme vaade, mis annab ülevaate ühest projekti või alamgrupi liikmest

Iga vaade koosneb komponentidest ning suures osas on need komponendid samad. Komponentidest on lähemalt jutt järgnevates alapeatükkides.

#### **4.3.1 Loetelu**

Iga ülevaate vaate esimeseks komponendiks on loetelu (Joonis 5). Vaate loetelu kohal on filtreerimislahter, mille abil saab valida, milliseid gruppe/projekte näidatakse loetelus, ning nupp “Update projects”, mis uuendab kasutaja kõiki selle vaate alla kuuluvaid projekte andmebaasis. Loetelus olevale grupile/projektile vajutades avaneb selle grupi/projekti vaade. Kui tegemist on grupi või alamgrupi vaatega, on lehe üleval paremas servas ka nupp “Change name”, mille abil saab grupi nime muuta. Grupi ja projekti eemaldamiseks on vastavalt nupp “Delete”.



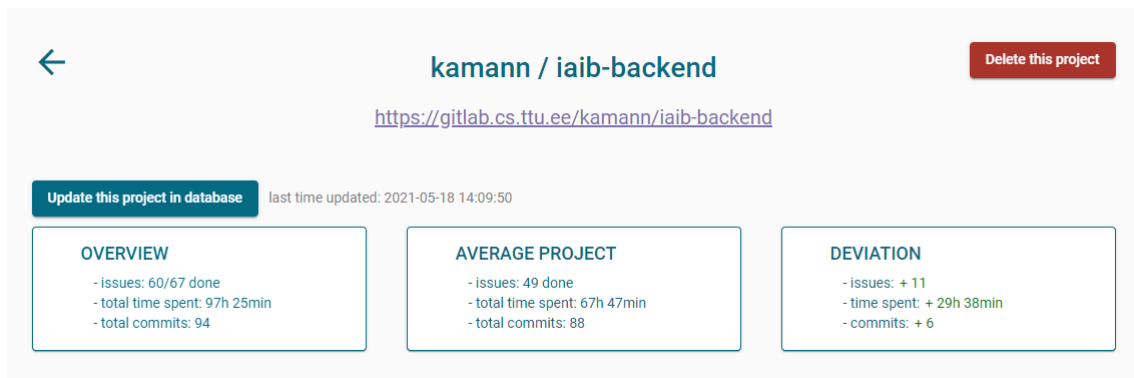
Joonis 5. Grupi loetelu vaade.

#### 4.3.2 Ülevaade ja võrdlus keskmisega

Ülevaade ja võrdlus keskmisega vaade (Joonis 6) on olemas alamgrupi, projekti ja liikme vaates. Alamgrupi ja projekti vaates on toodud välja *issue*'de koguarv, märgitud aeg ja nende *issue*'de arv, mis on staatuses “suletud” ning *commit*'ide koguarv, liikme vaates ainult märgitud aeg ning *commit*'ide arv.

Antud komponent koosneb kolmest ülevaadet andvast komponendist:

- ülevaade antud alamgrupi, projekti või liikme *issue*'de arvust, ajast ja/või *commit*'ide arvust
- ülevaade keskmise projekti *issue*'dest, ajast ja *commit*'ide arvust selles grupis või liikme puhul ülevaade keskmise liikme *commit*'ide arvust selles projektis
- antud alamgrupi, projekti või liikme andmete võrdlus keskmisega



Joonis 6. Projekti ülevaade ja võrdlus keskmisega vaade.

### 4.3.3 Liikmete tabel, filter ja liikme sidumine

Liikmete tabeli komponent (Joonis 7) on olemas alamgrupi ja projekti vaatel. Tabeli kohal on otsimise lahter, mis kuvab tabelisse uued andmed vastavalt valitud tähtpunktidele või ajavahemikule. Lisaks on komponendis nupud “Show this week data” ja “Show this month data”, mis kuvavad tabelisse vastavalt käesoleva nädala ja käesoleva kuu andmed. Tähtpunktide järgi andmete tabelisse kuvamine on võimalik vaid projekti vaates.

Tabelis endas on iga liikme kohta kirjas tema nimi, märgitud tööaeg ning *commit*’ide arv. Tabeli reale klikkimine viib edasi vastava liikme vaatesse.

Projekti vaates on lisaks liikmete tabelile tabeli all nime liikmega sidumise komponent. Kuna *commit*’id ei ole seotud GitLabi kasutajaga otse läbi id, vaid nime, on kasutajal vaja viia kokku niisugune nimi ja liige. Seda saab teha kahe sisendi välja kaudu. Esimeses on valikus kõik niisugused nimed, mis pole veel liikmega seotud. Teises on antud projekti kõik liikmed. Kui kasutaja on valinud vastava nime ja liikme, mida ta kokku soovib viia, peab ta vajutama nupule „PAIR“ ning seejärel salvestatakse nende seos andmebaasi. Peale nime kokku viimist liikmega on ka liikmete tabelis selle liikme *commit*’ide arv suurenenud, sest nüüd osatakse *commit*’i konkreetse nimega siduda. Juhul, kui ühtegi niisugust nime pole, mis poleks ühegi liikmega seotud, ei ole mainitud nime ja liikme sidumise komponenti näha.

**MEMBERS**

**Table filter**

**Filter by milestones**

Milestones ▼

**Filter by time**

Enter a date range 📅

Name	Username	Time	Commits
Kati Karu	kati	107h 35min	19
Mati Magu	mati	102h 48min	5
Andres Arkvel	andres	145h 35min	38

Some names were found that are not attached to any members. Feel free to pair them with a member from the project.

**Pick a name and the corresponding member**

Name

kati123 ▼

Member

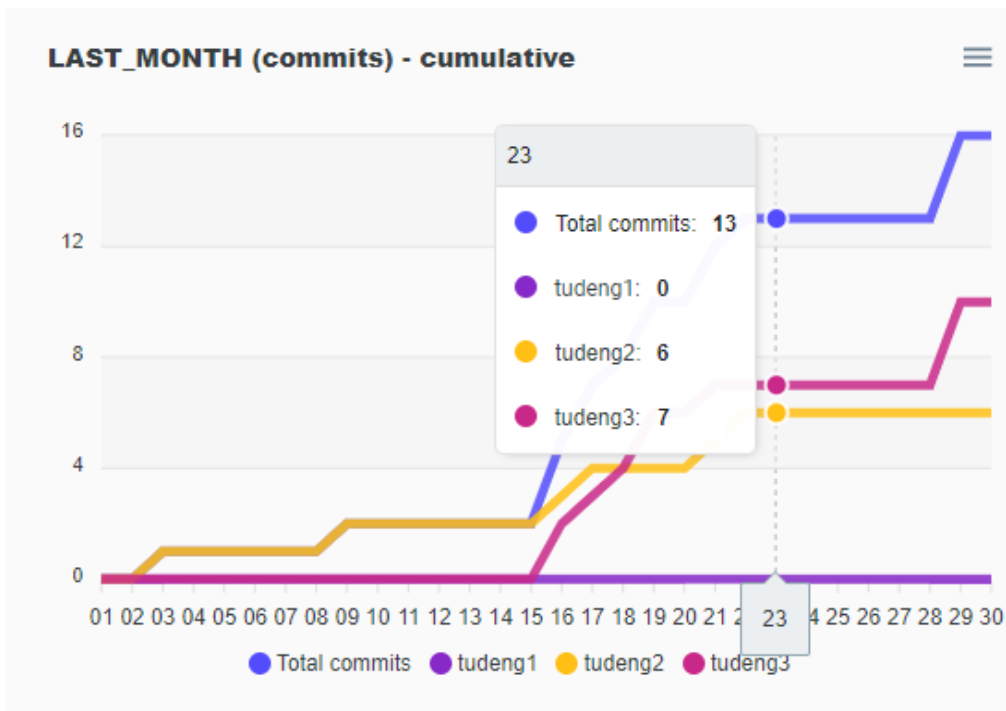
kati ▼

Joonis 7. Liikmete table ja liikme sidumise vaade.

#### 4.3.4 Graafikud ja otsimise lahter

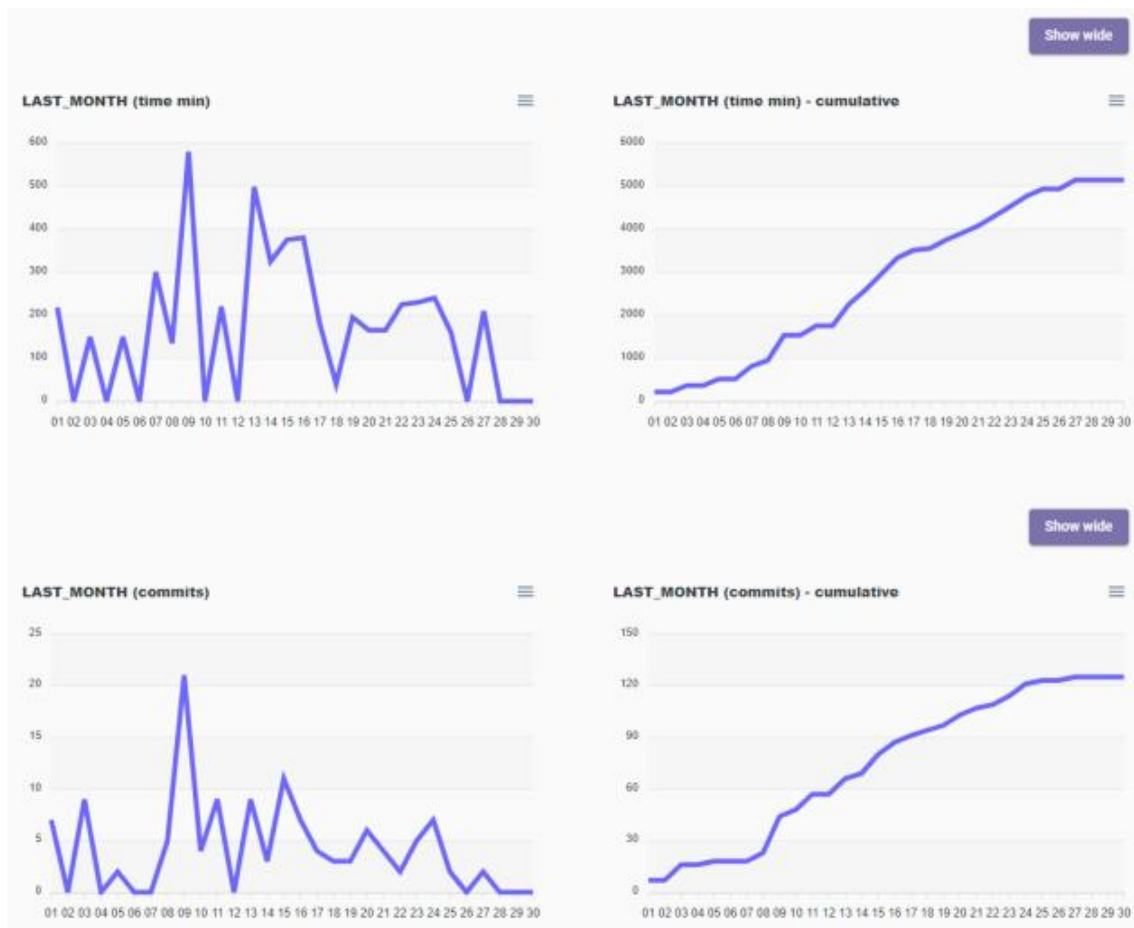
Graafikute komponent on igal ülevaate vaatel - gruppide, grupi, alamgrupi, projekti ja liikme vaatel. Graafikute kuvamine on antud töö peamine funktsionaalsus kasutaja vaatepunktist. Graafikutelt saab kõige paremini lugeda andmeid ning võrrelda erinevaid jooni graafikutes [4].

Et graafikutelt oleks informatsiooni kergem välja lugeda ja eristada, on graafikutes kasutatud palju erinevaid ja erksaid värve (Joonis 8).



Joonis 8. Graafiku detailvaade.

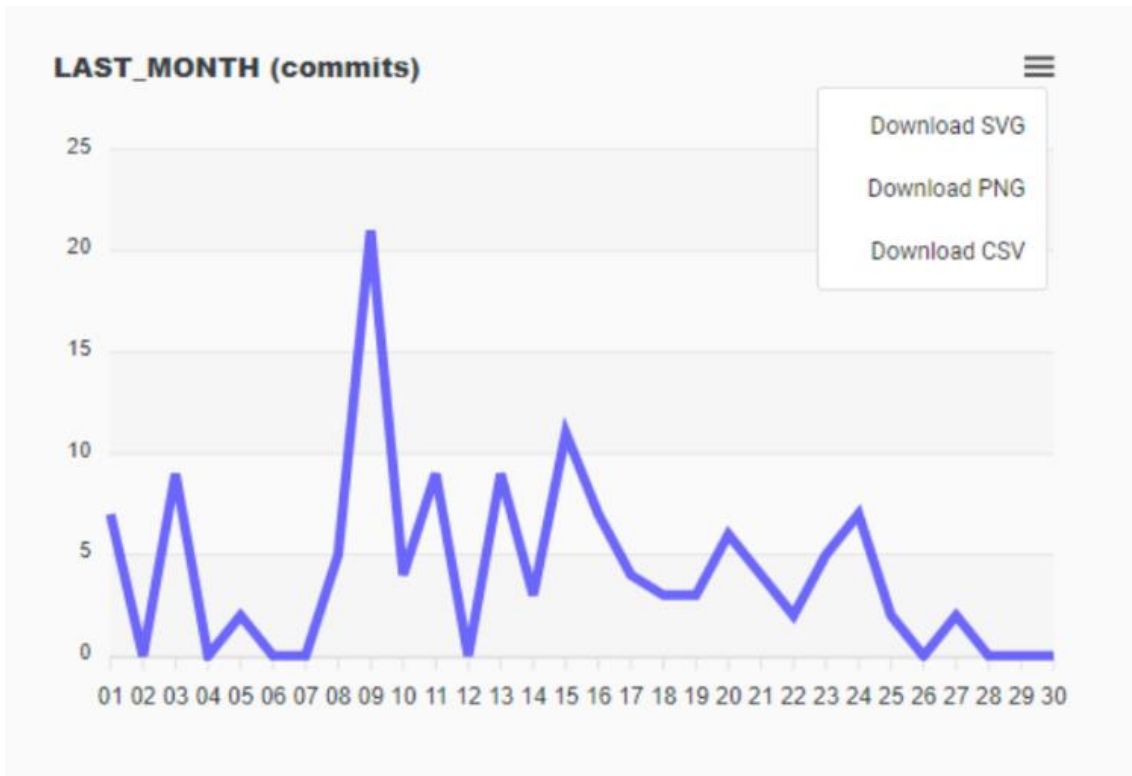
Graafiku komponendid on neljaosalised (Joonis 9) - graafik märgitud aja kohta, sama graafik kumulatiivsete andmetega ning graafik *commit*'ide arvu kohta ja sama graafik kumulatiivsete andmetega. *Commit*'ide graafikutel on liikme kohta arvestatud ainult nende *commit*'idega, millega ta seotud on. Siin tulevad taaskord mängu need lisanimed, mis *commit*'ide puhul võivad tekkida. Kui *commit*'i autorit pole võimalik kindlaks teha, ei kajastu see liikme *commit*'ide hulgas. Küll aga arvestatakse sellised *commit*'id projekti *commit*'ide hulka ning on graafiku kuvamisel sisse arvestatud. Graafikuid saab kuvada nii laiadena kui ka kitsastena, selleks on vastavalt nupud “Show wide” ja “Show narrow”.



Joonis 9. Neljaosaline graafikute komponent.

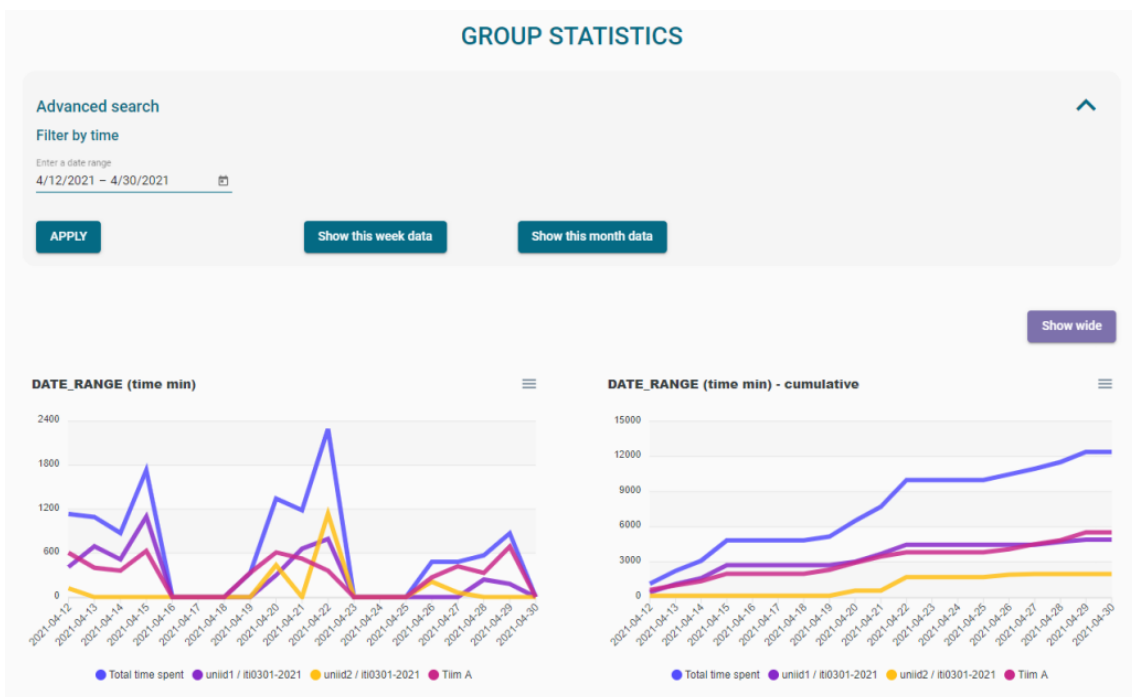
Vaikimisi on vaatel kolm graafiku komponenti - komponent eelmise nädala andmetega, eelmise kuu andmetega ning kogu aktiivse perioodi andmetega.

Iga graafiku juures on menüü, kust saab laadida alla selle graafiku SVG või PNG versiooni või andmed CSV-failina (Joonis 10).



Joonis 10. Graafiku vaade koos allalaadimisvõimalustega.

Graafikute kohal on otsimise lahter, mis näitab lisaks veel ühte neljagraafikulist graafikute komponenti. Otsida saab ajavahemike järgi ning lisaks on otsimisseksioonin nupud “Show this week data” ja “Show this month data”, mis kuvavad vastavalt lisagraafikutesse andmed käesoleva nädala või kuu kohta (Joonis 11).

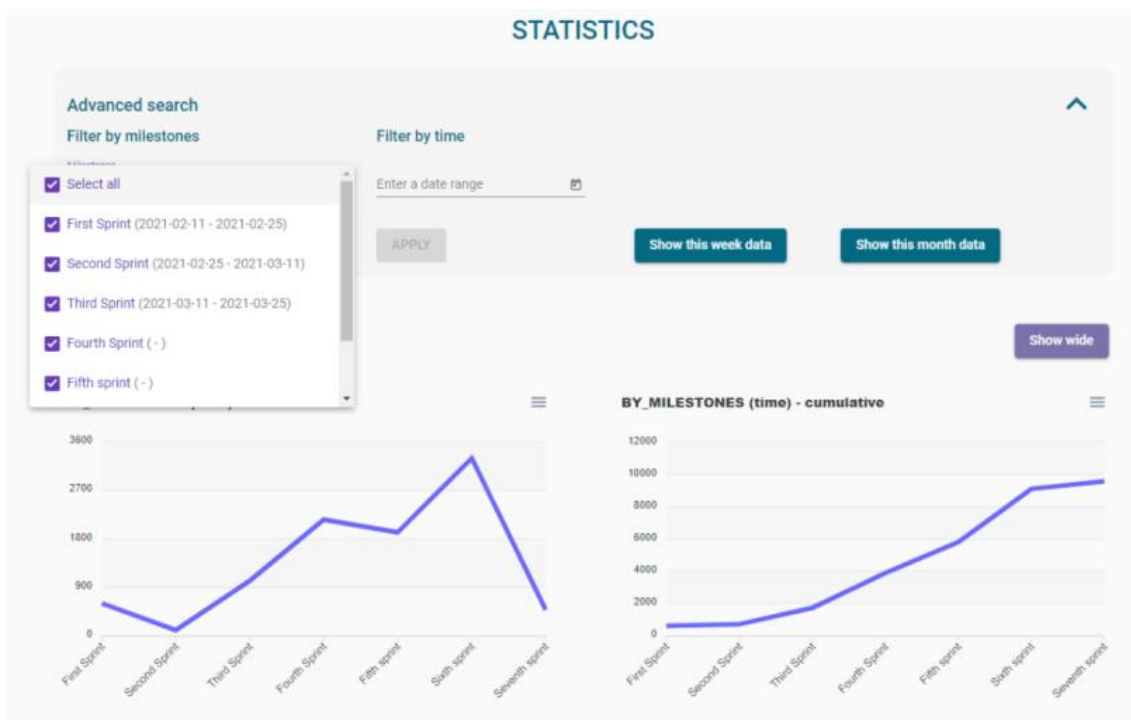


Joonis 11. Graafikute komponent koos otsimisseksiooniga.



Juhul, kui tegemist on projekti vaatega, saab graafikule kuvada andmeid ka tähtpunktide alusel. Tähtpunktidel saab olla nii algus- kui ka lõppkuupäev, kuid neid ei pea määrama. Kuna tähtpunktidel ei pruugi olla algus- ja lõppkuupäeva, siis ei saa tähtpunktide järgi otsingut teha nende algus- ja lõppkuupäevade järgi. Tähtpunktide alusel saab andmeid kuvada ainult *issue*'de järgi, kuna tähtpunktiga on otseselt seotud vaid *issue* (Joonis 12).

Märgitud aja kuvamisel tähtpunktide järgi võetakse tähtpunkti jaoks kokku kõikide selliste *issue*'de juurde märgitud aeg, mille tähtpunktiks on märgitud vastav tähtpunkt. *Commit*'ide puhul lähtutakse *commit*'i-sõnumist (ingl *commit message*). *Commit*'i sõnumis saab kirjutada, millise *issue*'ga antud *commit* seotud on. *Commit*'i sõnumisse kirjutatakse viide vastavale *issue*'le kujul #*issueId*, kus *issueId* on mainitud *issue* id. Samamoodi otsustatakse *issue*'ga seotud tähtpunkti järgi, et see *commit* arvestatakse antud tähtpunkti juurde. X-telje väärtusteks on valitud tähtpunktide nimed.




Joonis 12. Otsimisseksioon tähtpunktide järgi otsimise võimalusega.

#### 4.3.5 Kommentaari lahter

Kommentaari lahter on komponent, mis on olemas ainult projekti ja liikme vaatel. (Joonis 13). Kommentaari lahtrisse saab sisestada vaid ühe kommentaari, seda kommentaari saab iga aeg muuta. Kommentaar on näha ainult sellele kasutajale, kes selle kommentaari kirjutas. Juhul, kui ühel kasutajal on ühe GitLabi kasutaja kohta mitu projekti, siis on iga projekti alt valitud liikme kommentaar eraldiseisev. Sama kehtib projektide kohta

erinevates gruppides: kui üks projekt esineb mitmes grupis, siis on iga grupi alla kuuluva niisuguse projekti kommentaar eraldiseisev.



The image shows a user interface for leaving a comment. At the top, the word "COMMENT" is displayed in a bold, blue font. Below it, there is a link that says "Leave a comment". A text area contains the following text: "This team is in serious trouble because they don't know a first thing about coding. I think I should fail them. Just because I can. They'd anyway fail in their lives. Better make it quick. It's best for everyone if they just leave." To the right of the text area, there is a small icon of a pencil. Below the text area, there is a blue button with the word "SAVE" in white capital letters.

Joonis 13. Projekti/Liikme kommentaari vaade.

## 5 Kasutatud tehnoloogiad esirakenduses

Esirakenduseks ehk klientrakenduseks nimetatakse seda osa rakendusest, millega kasutaja suhtleb ja otseselt kokku puutub, sinna alla kuuluvad pildid, graafikud, tabelid, nupud, tekstid, värvid ja muu visuaalne info rakenduses. Esirakenduse arendamiseks kasutatavad keeled on HTML<sup>1</sup>, CSS<sup>2</sup> ja JavaScript<sup>3</sup> [5].

### 5.1 Angular

Angular on avatud lähtekoodiga TypeScriptipõhine veebiraamistik. TypeScript on avatud lähtekoodiga keel, mis ehitab end JavaScripti peale [6]. Angular koosneb komponentidest, iga komponendi osadeks on HTML mall, milles pannakse paika, mida lehele renderdatakse ja kuvatakse, TypeScripti klass, milles on loogika ning CSS valija (ingl *CSS selector*), mis defineerib, kuidas komponenti mallis kasutatakse [7]. Tänu suurele kasutajate hulgale on arendatud palju komponente, mida on võimalik igas Angulari rakenduses kasutada [8].

### 5.2 Apexcharts

ApexCharts<sup>4</sup> on avatud lähtekoodiga modernne graafikute koostamise teek (ingl *library*) [9]. ApexCharts võimaldab luua palju erinevaid interaktiivseid graafikuid. Et kasutada ApexCharts komponenti Angulari rakenduses, on Morris Janatzek teinud Angulari pakendi *ng-ApexCharts* [10] [11].

---

<sup>1</sup> <https://developer.mozilla.org/en-US/docs/Web/HTML>

<sup>2</sup> <https://www.w3.org/Style/CSS/Overview.en.html>

<sup>3</sup> <https://www.javascript.com/>

<sup>4</sup> <https://apexcharts.com/>

### 5.3 Angular Material

Angular Material<sup>1</sup> on kasutajaliidese komponentide teek, mis on Material Designi<sup>2</sup> avatud lähtekoodiga implementatsioon, mille on loonud ja mida haldab Google'i Angulari tiim. Angular Materialit saab kasutada Angulari projektides, et kiirendada elegantsete ja järjepidevate kasutajaliideste arendamist. Ta pakub taaskasutatavaid ja ilusaid kasutajaliidese komponente, millest osasid kasutab autor ka antud töös, nende hulgas näiteks Sidenav, Toolbar, Input, List, Divider, Progress spinner ja ikoonid [12] [13].

---

<sup>1</sup> <https://material.angular.io/>

<sup>2</sup> <https://material.io/develop/web>

## 6 Kasutatud tehnoloogiad tagarakenduses

Kui esirakendus viitab kasutajaliidesele, siis tagarakenduse ehk serverrakenduse alla kuuluvad server, rakendus ja andmebaas, mis töötavad tagaplaanil selle nimel, et kasutajale vajalikku informatsiooni edastada [14].

### 6.1 Spring Boot

Spring Boot on avatud lähtekoodiga Java<sup>1</sup> raamistik, mille abil on lihtne luua rakendusi. Spring Booti teeb muuhulgas mugavaks ja kergesti kasutatavaks see, et kõik on automaatselt seadistatud ning käsitsi pole vaja midagi seadistada. Lisaks lihtsustab Spring Boot sõltuvuste haldamist ning haldab REST-i lõpp-punkte [15].

Spring Security<sup>2</sup> on väga võimas ja kohandatav Java rakenduste autentimise raamistik, mis on de facto standardlahenduseks Springi rakendustele [16].

### 6.2 Gradle

Gradle<sup>3</sup> on avatud lähtekoodiga rakenduse ehitamise (ingl *build*) automatiseerimise süsteem, millega saab ehitada peaaegu igat tüüpi tarkvara. Gradle lihtsustab harilikku tüüpi projektide, näiteks Java teekide, ehitamist, lisades pistikprogrammide (ingl *plugin*) kaudu konventsioonide ja varasemalt valmis ehitatud funktsionaalsuse kihte [17].

### 6.3 MySQL andmebaas

MySQL on avatud lähtekoodiga andmebaasi haldamise süsteem, mis oli 2020 aasta detsembri seisuga üle kogu maailma populaarsuselt teine niisuguse süsteem [18]. MySQL

---

<sup>1</sup> <https://www.oracle.com/java/>

<sup>2</sup> <https://spring.io/projects/spring-security>

<sup>3</sup> <https://gradle.org/>

andmebaasi kasutavad mitmed maailma suurimad ja kiiremini arenevad organisatsioonid, nende hulgas Google ja Facebook [19].

## 7 Rakenduse võimalikud edasiarendused

Käesolevas peatükis tuuakse välja mõned võimalikud edasiarendused tulevikuks, mida töö koostamisel peeti võimalikuks ja mõistlikuks täiendavalt rakendusele lisada.

### 7.1 Kõik GitLabi domeenid

Antud rakendus on ehitatud töötama ainult TalTech GitLabiga, mille aadressiks on gitlab.cs.ttu.ee, kuid et veebilehte saaks kasutada laialdasemalt, võiks lisada kasutajale võimaluse valida, missugust GitLab domeeni ta kasutada soovib [20].

See nõuaks natuke andmebaasi muutmist, olenevalt sellest, kuidas see realiseerida, kuid on iseenesest üsna kerge muudatus. Kõige lihtsam viis seda realiseerida oleks võimaldada kasutajal määrata oma konto juures tema poolt kasutatav GitLabi URL-i algus. Sel juhul ei peaks muuta midagi muud peale tagarakenduses URL-i alguse andmebaasi salvestamise ja selle kasutamise. Niiviisi saaks kasutaja kasutada ühe kontoga vaid ühe GitLabi projekte. Et võimaldada kasutajal erinevate GitLabide projekte koos võrrelda, tuleks natuke teha ümber projekti salvestamist ning ligipääsuloa haldust – kasutaja peaks saama projekti lisades valida talle sobiliku GitLabi (näiteks gitlab.com) ning lisaks peaks laskma kasutajal siduda kontoga mitu GitLabi ligipääsuluba. Autoril jäi see selle lõputöö raames tegemata, sest see ei mahtunud skoopi ega olnud seatud eesmärkide täitmiseks vajalik.

### 7.2 Sisselogimine GitLabi kaudu

Kuna rakenduse funktsionaalseks kasutamiseks on vajalik, et kasutajal oleks GitLabi ligipääsuluba (ingl *access token*), siis järelikult peaks kasutajal olema ka GitLabi kasutaja. Et teha veebilehele sisselogimine mugavamaks, võiks lisada võimaluse logida sisse GitLabi kasutajaga. Niimoodi toimides saaks ka ligipääsuloa küsimise kasutajalt endalt vahele jätta ning kasutada otse GitLabi konto küljest seda ligipääsuluba.

### **7.3 Jagamine**

Kuigi veebirakendus on mõeldud kasutamiseks eelkõige õppejõule tudengite tööde mahtude võrdlemiseks, siis oleks kena, kui kasutajal oleks võimalus jagada kas kogu grupi/projekti ülevaadet või ainult enda valitud osa sellest teiste kasutajatega, kelleks oleks hüpoteetiliselt tudengid, et ka nemad näeksid enda projekti ja tööd võrdluses teiste projektidega.



## 8 Kokkuvõte

Antud töö eesmärgiks oli luua veebipõhine rakendus, mis annaks ülevaate TalTechi GitLabi projektidest ning aitaks võrrelda nii projekte omavahel kui ka projekti liikmete panust. Töös oli toodud välja rakenduse funktsionaalsed nõuded, analüüsitud olemasolevaid sarnaseid rakendusi, kirjeldatud valminud rakenduse funktsionaalsust ja vaateid.

Valminud veebileht on alles esimene versioon ning seda saab vajaduse ja tahtmise korral edasi arendada. Funktsionaalsusest saaks lisada toe erinevate GitLabi domeenide jaoks, GitLabi sisselogimise ning gruppide/projektide ülevaate jagamise mingil moel teiste rakenduse kasutajatega.

## Kasutatud kirjandus

- [1] „Is it any good?“, GitLab, [Võrgumaterjal]. Saadaval: <https://about.gitlab.com/is-it-any-good/>. [Kasutatud 20 mai 2021].
- [2] A. Prakash, „Top GitHub Alternatives to Host Your Open Source Projects,“ It's FOSS, 15 12 2020. [Võrgumaterjal]. Saadaval: <https://itsfoss.com/github-alternatives/>. [Kasutatud 20 mai 2021].
- [3] „What is an API Endpoint?“, Smartbear, [Võrgumaterjal]. Saadaval: <https://smartbear.com/learn/performance-monitoring/api-endpoints/>. [Kasutatud 16 mai 2021].
- [4] V. Pacha, „Line Graph - Definition, Types, Examples | How to Construct a Line Graph?“, CCSSMathAnswers, 19 aprill 2021. [Võrgumaterjal]. Saadaval: <https://ccssmathanswers.com/how-to-construct-a-line-graph/>. [Kasutatud 24 05 2021].
- [5] palaksinghal9903, „Frontend vs Backend,“ GeeksforGeeks, 09 03 2021. [Võrgumaterjal]. Saadaval: <https://www.geeksforgeeks.org/frontend-vs-backend/>. [Kasutatud 16 mai 2021].
- [6] „What is TypeScript?“, TypeScript, [Võrgumaterjal]. Saadaval: <https://www.typescriptlang.org/>. [Kasutatud 16 mai 2021].
- [7] „Angular,“ Angular, [Võrgumaterjal]. Saadaval: <https://angular.io/>. [Kasutatud 16 mai 2021].
- [8] „Angular Review,“ Slant, [Võrgumaterjal]. Saadaval: <https://www.slant.co/options/14704/~angular-review>. [Kasutatud 16 mai 2021].
- [9] „ApexCharts - Reviews, Pros & Cons,“ Stackshare, [Võrgumaterjal]. Saadaval: <https://stackshare.io/apexcharts>. [Kasutatud 16 mai 2021].
- [10] „ApehCharts.js,“ APEXCHARTS, [Võrgumaterjal]. Saadaval: <https://apexcharts.com/>. [Kasutatud 16 mai 2021].
- [11] „GitHub,“ [Võrgumaterjal]. Saadaval: <https://github.com/apexcharts/ng-apexcharts>. [Kasutatud 16 mai 2021].
- [12] O. Mensah, „Creating Beautiful Apps with Angular Material,“ 14 09 2019. [Võrgumaterjal]. Saadaval: <https://auth0.com/blog/creating-beautiful-apps-with-angular-material/>. [Kasutatud 16 mai 2021].
- [13] „Material Design,“ [Võrgumaterjal]. Saadaval: <https://material.io/develop/web>. [Kasutatud 16 mai 2021].
- [14] „What Is The Difference Between Front-End And Back-End Developer,“ Concepta, 27 02 2019. [Võrgumaterjal]. Saadaval: <https://www.conceptatech.com/blog/difference-front-end-back-end-development>. [Kasutatud 16 mai 2021].

- [15] „Spring Boot - Introduction,“ Tutorialspoint, [Võrgumaterjal]. Saadaval: [https://www.tutorialspoint.com/spring\\_boot/spring\\_boot\\_introduction.htm](https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm). [Kasutatud 16 mai 2021].
- [16] „Spring Security,“ Soring, [Võrgumaterjal]. Saadaval: <https://spring.io/projects/spring-security>. [Kasutatud 16 mai 2021].
- [17] „What is Gradle?,“ Gradle, [Võrgumaterjal]. Saadaval: [https://docs.gradle.org/current/userguide/what\\_is\\_gradle.html](https://docs.gradle.org/current/userguide/what_is_gradle.html). [Kasutatud 16 mai 2021].
- [18] „Ranking of the most popular database management systems worldwide, as of December 2020\*,“ Statista, 2021. [Võrgumaterjal]. Saadaval: <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/>. [Kasutatud 16 mai 2021].
- [19] „Why MySQL?,“ MySQL, [Võrgumaterjal]. Saadaval: <https://www.mysql.com/why-mysql/>. [Kasutatud 16 mai 2021].
- [20] „Custom domains and SSL/TLS Certificates,“ GitLab, [Võrgumaterjal]. Saadaval: [https://docs.gitlab.com/ee/user/project/pages/custom\\_domains\\_ssl\\_tls\\_certification/](https://docs.gitlab.com/ee/user/project/pages/custom_domains_ssl_tls_certification/). [Kasutatud 16 mai 2021].

## **Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks<sup>1</sup>**

Mina, Kadri Männimets

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Veebirakendus TalTechi GitLabi grupiprojektidest ülevaate saamiseks ja nende võrdlemiseks“, mille juhendaja on Ago Luberg
  - 1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

16.05.2021

---

<sup>1</sup> Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingu tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktile 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

## Lisa 2 – Andmebaasiskeem



## Lisa 3 – GitLab API projekti päringu vastuse näide

```
{
  "id": 17767,
  "description": "",
  "name": "iaib",
  "name_with_namespace": "kamann / iaib",
  "path": "iaib",
  "path_with_namespace": "kamann/iaib",
  "created_at": "2021-02-02T14:49:48.054Z",
  "default_branch": "master",
  "tag_list": [],
  "ssh_url_to_repo":
"git@gitlab.cs.ttu.ee:kamann/iaib.git",
  "http_url_to_repo":
"https://gitlab.cs.ttu.ee/kamann/iaib.git",
  "web_url": "https://gitlab.cs.ttu.ee/kamann/iaib",
  "readme_url": "https://gitlab.cs.ttu.ee/kamann/iaib/-
/blob/master/README.md",
  "avatar_url": null,
  "forks_count": 0,
  "star_count": 0,
  "last_activity_at": "2021-05-09T16:47:15.865Z",
  "namespace": {
    "id": 1340,
    "name": "kamann",
    "path": "kamann",
    "kind": "user",
    "full_path": "kamann",
    "parent_id": null,
    "avatar_url": null,
    "web_url": "https://gitlab.cs.ttu.ee/kamann"
  },
  "_links": {
    "self":
"https://gitlab.cs.ttu.ee/api/v4/projects/17767",
    "issues":
"https://gitlab.cs.ttu.ee/api/v4/projects/17767/issues",
    "merge_requests":
"https://gitlab.cs.ttu.ee/api/v4/projects/17767/merge_reque
sts",
    "repo_branches":
"https://gitlab.cs.ttu.ee/api/v4/projects/17767/repository/
branches",
    "labels":
"https://gitlab.cs.ttu.ee/api/v4/projects/17767/labels",
    "events":
"https://gitlab.cs.ttu.ee/api/v4/projects/17767/events",
    "members":
"https://gitlab.cs.ttu.ee/api/v4/projects/17767/members"
```

```

},
"packages_enabled": true,
"empty_repo": false,
"archived": false,
"visibility": "private",
"owner": {
  "id": 1291,
  "name": "kamann",
  "username": "kamann",
  "state": "active",
  "avatar_url": null,
  "web_url": "https://gitlab.cs.ttu.ee/kamann"
},
"resolve_outdated_diff_discussions": false,
"container_registry_enabled": true,
"container_expiration_policy": {
  "cadence": "1d",
  "enabled": false,
  "keep_n": 10,
  "older_than": "90d",
  "name_regex": ".*",
  "name_regex_keep": null,
  "next_run_at": "2021-02-03T14:49:48.068Z"
},
"issues_enabled": true,
"merge_requests_enabled": true,
"wiki_enabled": true,
"jobs_enabled": true,
"snippets_enabled": true,
"service_desk_enabled": false,
"service_desk_address": null,
"can_create_merge_request_in": true,
"issues_access_level": "enabled",
"repository_access_level": "enabled",
"merge_requests_access_level": "enabled",
"forking_access_level": "enabled",
"wiki_access_level": "enabled",
"builds_access_level": "enabled",
"snippets_access_level": "enabled",
"pages_access_level": "private",
"operations_access_level": "enabled",
"analytics_access_level": "enabled",
"emails_disabled": null,
"shared_runners_enabled": true,
"lfs_enabled": true,
"creator_id": 1291,
"import_status": "none",
"import_error": null,
"open_issues_count": 4,
"runners_token": "Xdq5cXbDAMP2YyusU6Zy",
"ci_default_git_depth": 50,

```

```

"ci_forward_deployment_enabled": true,
"public_jobs": true,
"build_git_strategy": "fetch",
"build_timeout": 3600,
"auto_cancel_pending_pipelines": "enabled",
"build_coverage_regex": null,
"ci_config_path": null,
"shared_with_groups": [],
"only_allow_merge_if_pipeline_succeeds": false,
"allow_merge_on_skipped_pipeline": null,
"restrict_user_defined_variables": false,
"request_access_enabled": true,
"only_allow_merge_if_all_discussions_are_resolved":
false,
"remove_source_branch_after_merge": true,
"printing_merge_request_link_enabled": true,
"merge_method": "merge",
"suggestion_commit_message": null,
"auto_devops_enabled": false,
"auto_devops_deploy_strategy": "continuous",
"autoclose_referenced_issues": true,
"permissions": {
  "project_access": {
    "access_level": 40,
    "notification_level": 0
  },
  "group_access": null
}
}

```



## Lisa 4 – GitLab API projekti *issue* märkmete päringu ühe märkme vastuse näide

```
{
  "id": 137489,
  "type": null,
  "body": "added 448mo 2w 3h 18m 38s of time spent at 2021-04-20",
  "attachment": null,
  "author": {
    "id": 1291,
    "name": "kamann",
    "username": "kamann",
    "state": "active",
    "avatar_url": null,
    "web_url": "https://gitlab.cs.ttu.ee/kamann"
  },
  "created_at": "2021-04-20T10:10:55.041Z",
  "updated_at": "2021-04-20T10:10:55.044Z",
  "system": true,
  "noteable_id": 21780,
  "noteable_type": "Issue",
  "resolvable": false,
  "confidential": false,
  "noteable_iid": 82,
  "commands_changes": {}
}
```