

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**INFORMATIKOS FAKULTETAS**

**P175B015 Programų sistemų inžinerija**

**„To-Do“ darbų užrašinė**

Atliko: Augustas Maslauskas IFF-6/11

Paulius Daščioras IFF-6/11

Linas Raičinskis IFF-6/11

Vilius Kerutis IFF-6/7

**KAUNAS 2018**

[1. Įvadas 3](#_Toc509090410)

[2. Analizė 4](#_Toc509090411)

[2.1. Projekto įvadas 4](#_Toc509090412)

[2.2. Techninis pasiūlymas (algoritmas) ir technologijos 4](#_Toc509090413)

[2.3. Funkciniai ir nefunkciniai projekto reikalavimai 4](#_Toc509090414)

[2.4. Puslapio pirminė išvaizda bei vartotojo sąsaja (*User Interface*) 5](#_Toc509090420)

[2.5. Projekto *Backlog screenshot‘as* bei trumpas aprašas 8](#_Toc509090421)

[2.6. Nuoroda į kodo saugyklą (naudojamą *repo*) trumpas šakų (*branches* aprašymas) 9](#_Toc509090422)

# Įvadas

Šio projekto tikslas yra įsisavinti vieną iš projektavimo metodikų: *Scrum* bei taikant ją, realizuoti funkcionuojančią sistemą. Nusprendėme kurti android aplikaciją, kuri padėtų vartotojams lengviau planuoti savo laiką, tinkamai paskirstyti užduotis, paskatintų užduočių vykdymui skirti laiko kiekvieną dieną. Šiam projektui iškėlėme tokius reikalavimus:

1. Naudoti *Scrum* programavimo metodiką
2. Aprašyti sistemos dokumentaciją
3. Sistema turi išpildyti funkcinius ir nefunkcinius reikalavimus
4. Kiekvienam sprintui iškeliami tikslai

Šį darbą sudaro *backlog‘o* sukūrimas, užduočių vykdymo laiko kiekvieną dieną algoritmo sukūrimas, duomenų bazės sukūrimas ir sujungimas, sistemos testavimas.

# Analizė

## Projekto įvadas

„To-Do“ tikslas supaprastinti užduočių atlikimo laiko paskirstymą. Žmogus suvedęs užduotis turės galimybę matyti, kiek laiko per dieną turi skirti užduočių atlikimui. Jeigu per dieną nėra skiriama laiko, tai kitom dienom užduočių atlikimo laikas prasiilgina. Atėjus suplanuotam užduoties atlikimo laikui, naudotojas gauna pranešimą, kad reikia pradėti užduotį, su dviem pasirinkimais: pradėti, vėliau. Jei užduotis buvo pradėta, praėjus suplanuotam laikui ekrane pasirodo lentelė su dviem mygtukais: baigti, tęsti.

## Techninis pasiūlymas (algoritmas) ir technologijos

Sistemai sukurti naudosime:

1. *Back-end* programavimui: JAVA SE 9 versija. Pasirinkome šią programavimo kalbą, nes ši kalba mums patogiausia kuriant android aplikaciją.
2. *Front-end* programavimui: *XML 1.0*.
3. Duomenų bazė: *MYSQL*. Rinkomės tarp *MYSQL* ir *MSSQL*, pasirinkome MYSQL duomenų bazę, nes turime patirties su ja. Taip pat duomenų bazei sujungti naudosime *phpMyAdmin* ir *SQL* serverį.

Algoritmas – rašysime savo algoritmą, nes neradome nieko panašaus, ko mums reikia. Algoritmo veikimas: Algoritmas, iš nurodyto užduoties preliminaraus atlikimo laiko, paskirsto konkrečiom valandom skirtą laiką. Taip pat, jeigu vartotojas negali tą dieną atlikti užduoties arba nutęsia užduotį, jis automatiškai perskirsto laiką.

## Funkciniai ir nefunkciniai projekto reikalavimai

Funkciniai reikalavimai:

## Naujo vartotojo registracija.

## Užduoties sukūrimas.

## Užduočių peržiūra.

## Kategorijos kūrimas.

## Vartotojo prisijungimas.

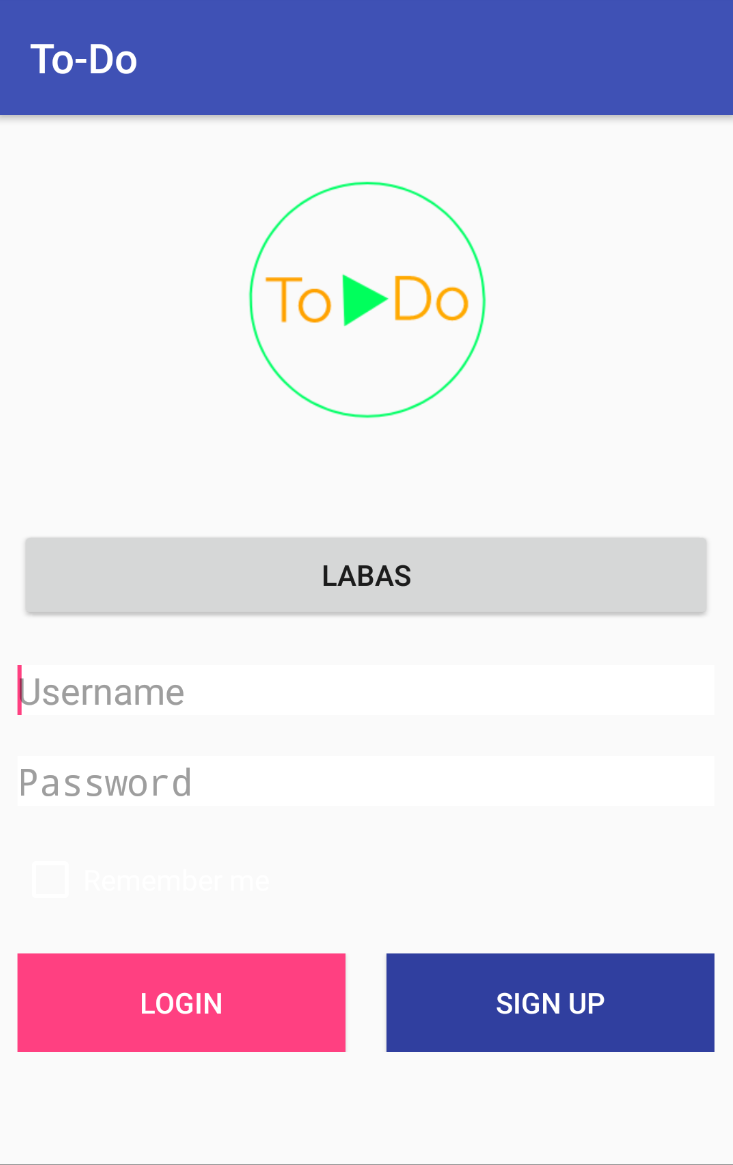
1. Kitų vartotojų pridėjimas į kategoriją.
2. Užduoties pradėjimas arba atidėjimas gavus priminimą.

Nefunkciniai reikalavimai:

1. Aplikacija palaiko andoid versijas nuo “Ice Cream Sandwich” ir naujesnes.
2. Aplikacija prisitaiko prie skirtingų ekranų rezoliucijų.

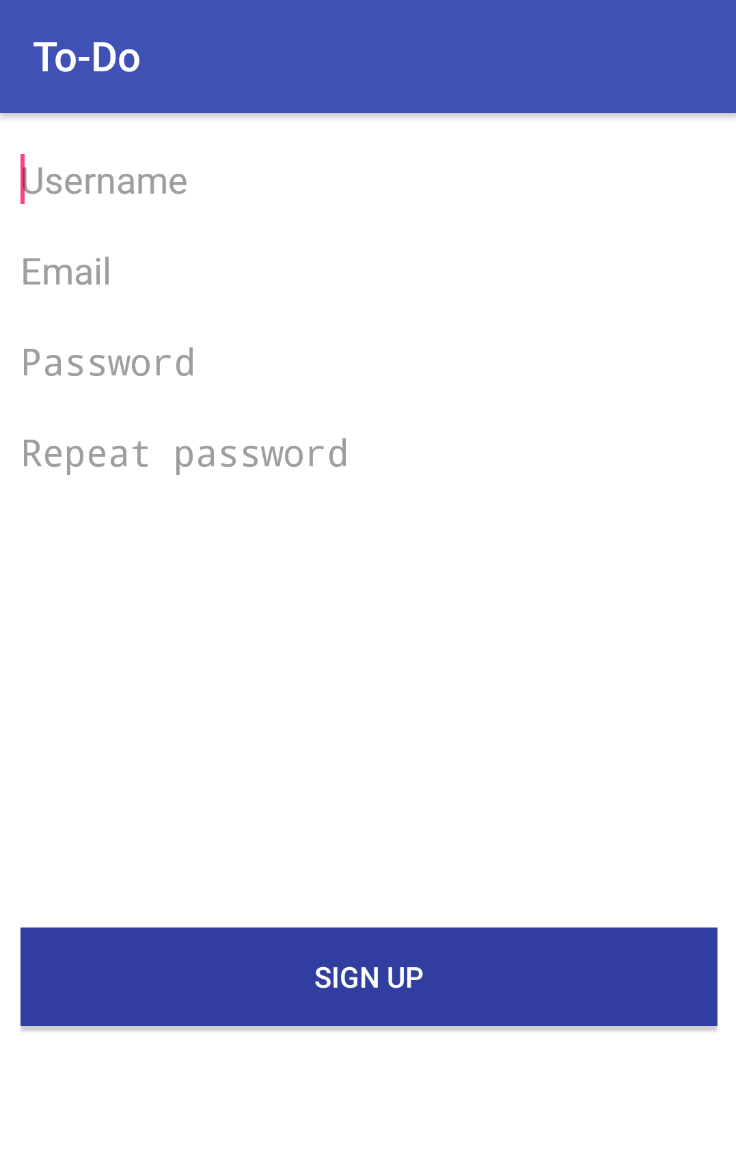
## Puslapio pirminė išvaizda bei vartotojo sąsaja (*User Interface*)

Pradinis puslapis:



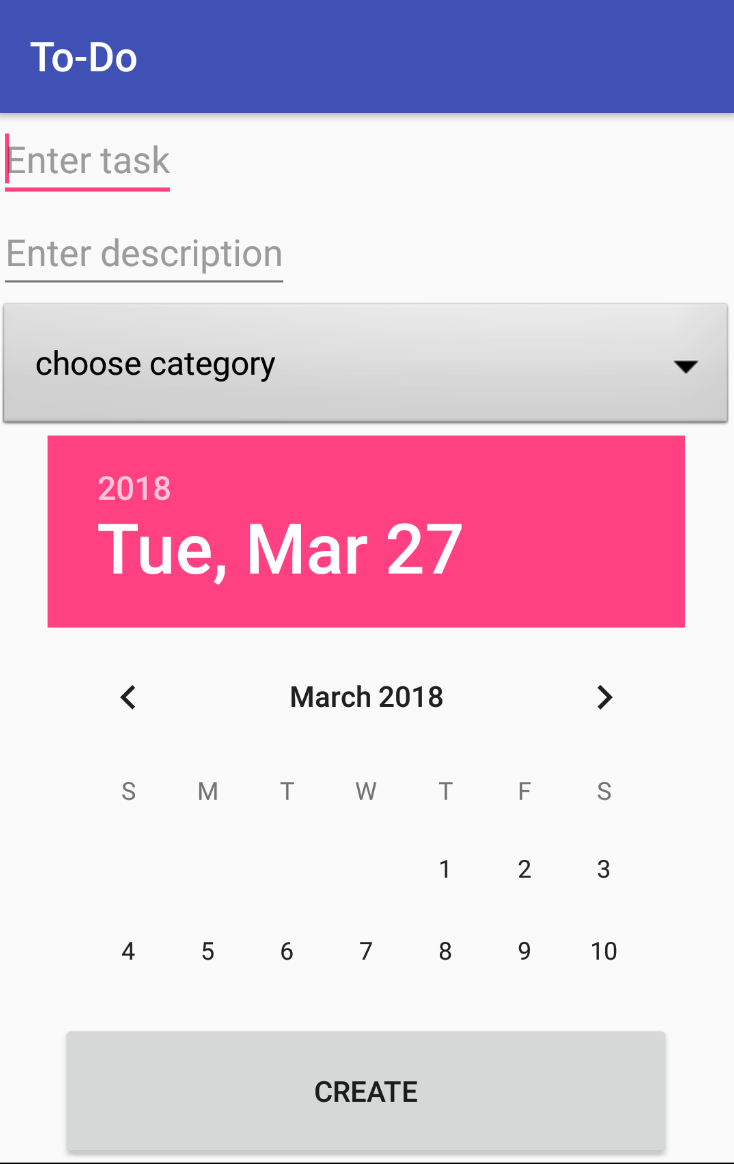
1 pav. Pradinio lango (Welcome page) atvaizdis.

Registracijos forma:



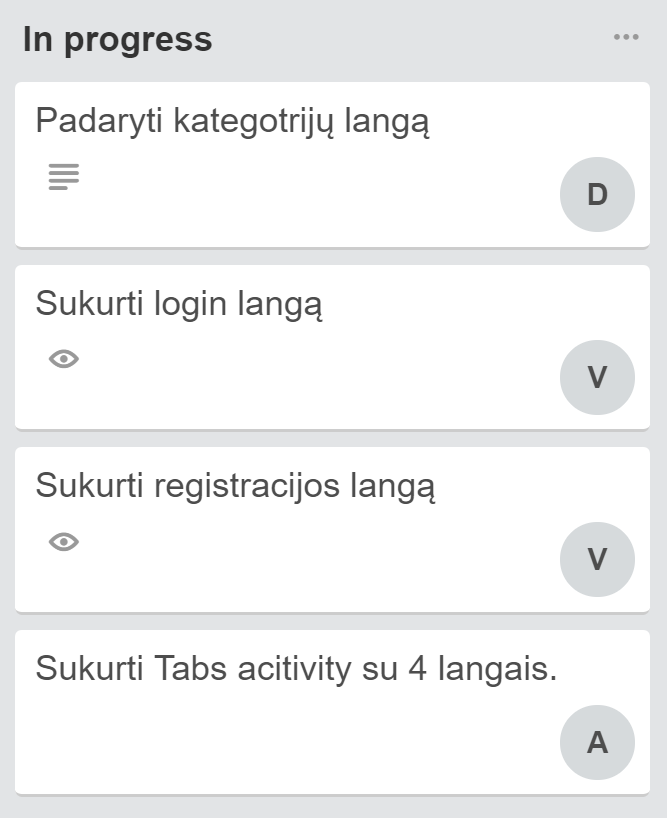
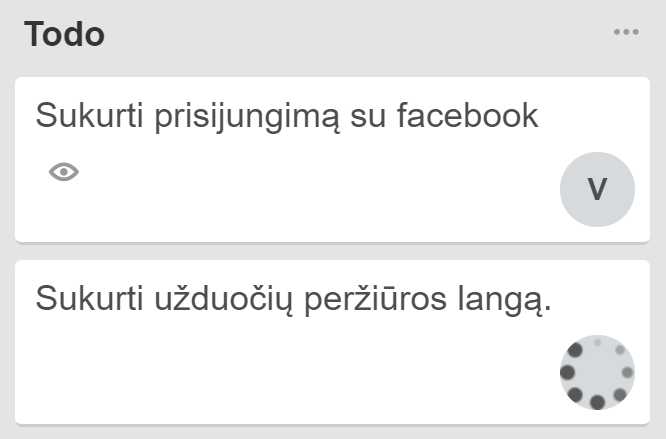
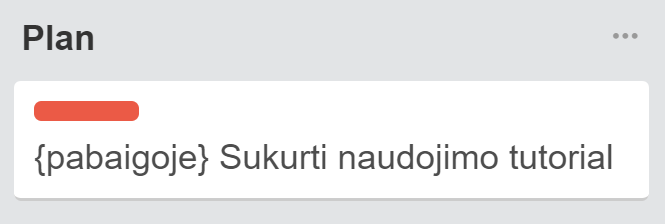
2 pav. Registracijos lango advaizdis.

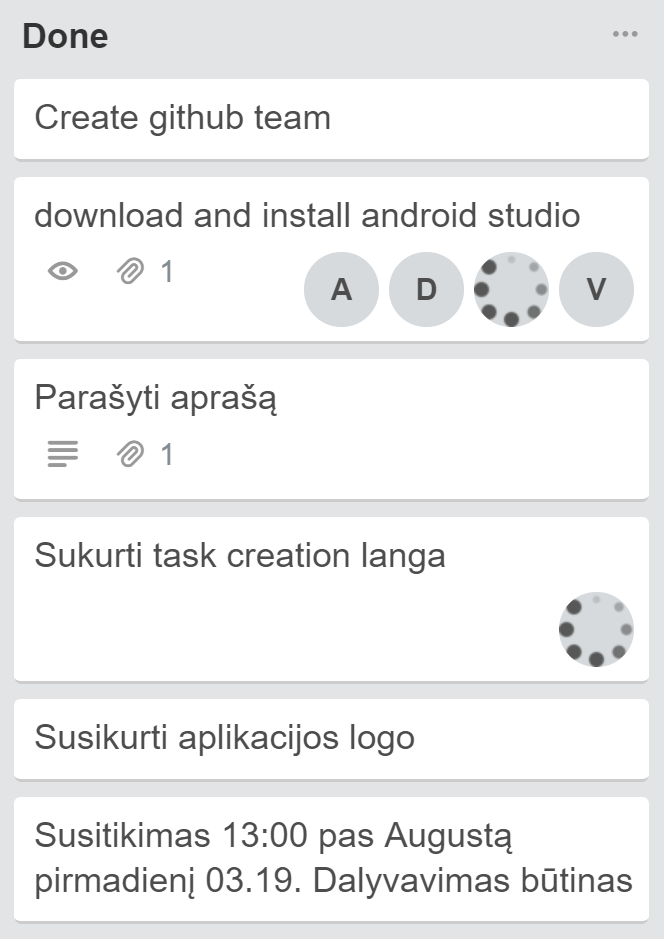
Užduoties kūrimo forma:



3 pav. Užduoties kūrimo lango advaizdis.

## Projekto *Backlog screenshot‘as* bei trumpas aprašas





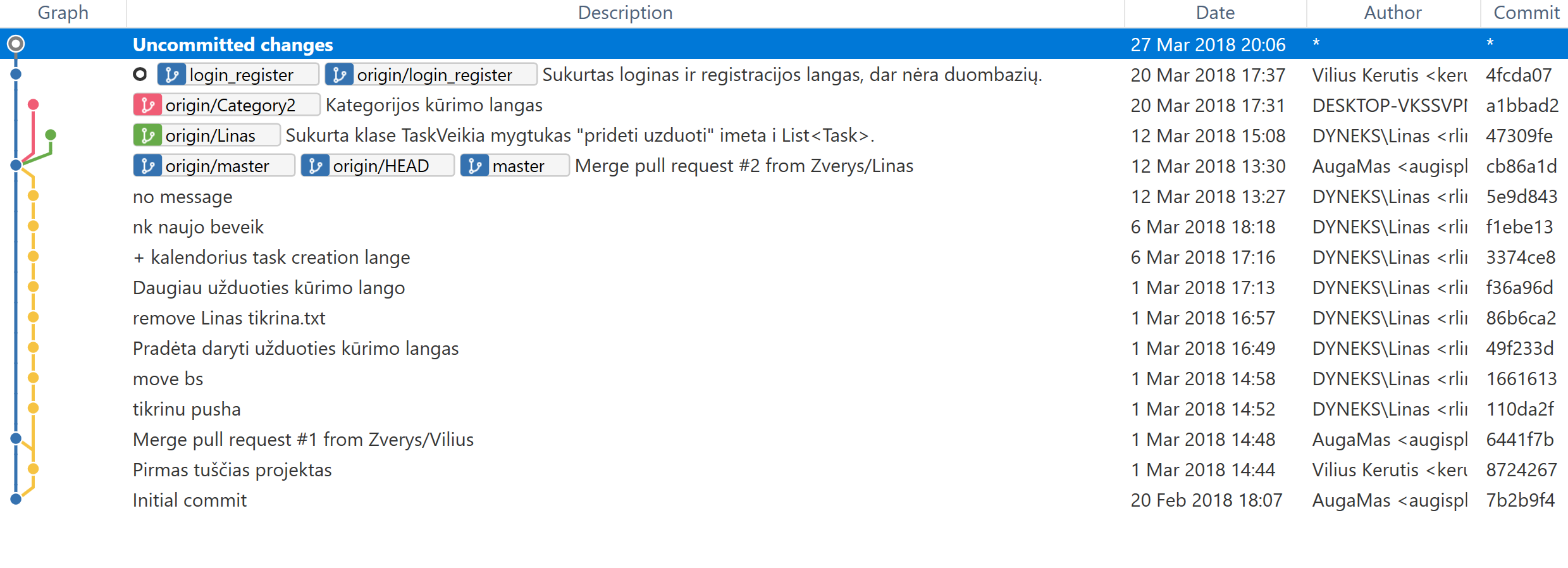
4 pav. “Trello” programoje sudarytas užduočių sąrašas

Atsižvelgiant į mūsų *backlog‘ą*, galima iškelti šiuos svarbiausius projekto uždavinius:

1. Sukurti login langą, kuris kontaktuotų su duombaze, taip pat sukurti dizainą šiam langui.
2. Sukurti prisijungimą su facebook.
3. Kad sukurti lagin langą, savaime aišku, reikia turėti registracijos langą, taigi jį sukurti būtina, jis turės kontaktuoti su duombaze.
4. Sukurti kategorijų kūrimo langą. Tai svarbu, nes vėliau bus kuriamos užduotys, kuriose reikės pasirinkti, kokiai kategorijai priklauso užduotis.
5. Sukurti „TabsActivity“, kuris reikalingas slinkimui(angl. swipe) nuo vieno lango prie kito. Tai viena iš svarbiausių projekto detalių, nes komanda nori pasipraktikuoti šioje aplikacijų kūrimo srityje.
6. Tolimas tiklsas: po visko sukurti naudojimosi programa vadovą. Planuojamas aprašas vartotojui, kuris ketina naudotis mūsų sukurtu projektu.

## Nuoroda į kodo saugyklą (naudojamą *repo*) trumpas šakų (*branches*) aprašymas

* + 1. Nuoroda į kodo saugyklą: <https://github.com/orgs/Zverys/dashboard>
    2. Projekte naudosime pagrindinę(master) šaką, iš kurios išeina „develop“ šaka. Kiekvienai projekte naudojamai funkcijai(angl. feature) sukurti bus kuriama nauja šaka pavadinimu „/feature/[kuriamos funkcijos pavadinimas]“. Sukurtos funkcijos bus ištestuotos ir atgal apjungiamos su „master“ šaka;
    3. Sujungimas į „master“ šaką įvyks po kiekvieno „sprinto“.



5 pav “SourceTree” programoje vaizduojamas medis

## Projekto testavimas

Testuodami projekto funkcionalumą ir veikimą, atlikę programos kodo pakeitimus, naudosime unit testus. Užtikrinant programos veikimą be jokių klaidų atliksime ir integracinius testus. Šiais dviem testavimo tipais užtikrinsime visų projekto funkcijų tinkamą ir stabilų veikimą, naudojantis programa. Taip pat testuodami naudosime Mock įrankį, kuris padės mums testuoti, pavyzdžiui, kvietimus su duomenų baze.