

# Projektų užduočių valdymo programa

Versija 0.2

<b>Projektų užduočių valdymo programa</b>	1
<b>Apie sistemą</b>	1
<b>Funkciniai reikalavimai</b>	2
FR0. Vartotojai gali prisijungti prie sistemos *	2
Sistemoje yra galimybė prisijungti ir atsijungti. Prisijungus matomas vartotojo vardas.	2
FR1. Sistemoje pateikiamas projektų sąrašas	2
FR2. Sistema leidžia administruoti projektus	2
FR3. Projekte pateikiamas užduočių sąrašas	2
FR4. Projekte egzistuoja galimybė administruoti užduotis	2
FR5. Projekte egzistuoja užduočių būsenų lenta	3
FR6. Sistemoje egzistuoja galimybė ieškoti projektų ir užduočių	3
Projektų paieška turi būti vykdoma pagal šiuos kriterijus:	3
Užduočių paieška turi būti vykdoma pagal šiuos kriterijus:	3
FR7. Projektų sąrašas gali būti eksportuojamas CSV formatu.	3
FR8. Užduočių sąrašas gali būti eksportuojamas CSV formatu.	3
<b>Nefunkciniai reikalavimai</b>	3
NFR1. Sistema įgyvendinta naudojant šias technologijas:	3
NFR2. Vartotojo sąsajai įgyvendinti naudoti atitinkamai:	4
NFR3. Sistemos duomenys saugomi reliacinėje duomenų bazėje	4
NFR4. Visi veiksmai sistemoje kaupiami įvykių žurnale	4
NFR5. Kartu su sistema pateikiama sistemos diegimo ir naudojimo dokumentacijos	4
NFR6. Visi veiksmai sistemoje turi atsakyti greičiau nei per 2 sekundes.	4

## Apie sistemą

Projektų užduočių valdymo sistema skirta palaikyti vykdomų projektų procesą. Įmonei, įsidiegusiai programinę įrangą, suteikiama galimybė registruoti projektus ir sekti jų įgyvendinimą.

# Funkciniai reikalavimai

## FR0. Vartotojai gali prisijungti prie sistemos \*

Sistemoje yra galimybė prisijungti ir atsijungti. Prisijungus matomas vartotojo vardas.

## FR1. Sistemoje pateikiamas projektų sąrašas

Sistemos pagrindiniame lange pateikiamas projektų sąrašas. Projekto sąrašas turi teikti tokią informaciją:

- Projekto pavadinimas
- Projekto aprašymas
- Projekto informacija:
  - Projekto būseną: *vykdomas, užbaigtas*
  - Projekto bendras užduočių kiekis
  - Projekto neatliktų užduočių kiekis

## FR2. Sistema leidžia administruoti projektus

Sistemoje turi būti numatytas:

- Projekto kūrimas
- Projektų trynimasis
- Projekto nustatymų redagavimas: pagal FR1

## FR3. Projekte pateikiamas užduočių sąrašas

Projekto lange pateikiamas užduočių sąrašas. Užduočių sąrašas turi teikti tokią informaciją:

- Užduoties ID
- Užduoties pavadinimas
- Užduoties aprašymas (User story formatas)
- Užduoties prioritetas: žemas, vidutinis, aukštas
- Užduoties būseną: padaryti, daroma, padaryta
- Automatiškai užpildomos užduoties sukūrimo ir atnaujinimo datos.

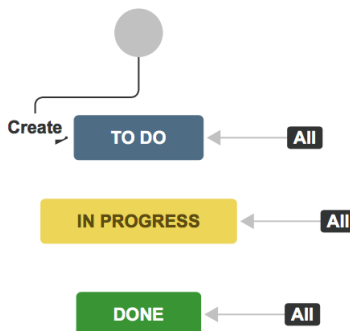
## FR4. Projekte egzistuoja galimybė administruoti užduotis

Projekte turi būti numatytas:

- Užduoties kūrimas
- Užduoties trynimasis
- Užduoties nustatymų redagavimas: pagal FR3

## FR5. Projekte egzistuoja užduočių būsenų lenta

Projekte egzistuoja užduočių būsenų lenta, kurioje turi būti matomos užduotys ir jų būsenos. Lentoje turi būti 3 stulpeliai: TODO, IN PROGRESS, DONE. Lenta turi atvaizduoti pateiktą užduočių srautą:



## FR6. Sistemoje egzistuoja galimybė ieškoti projektų ir užduočių

Projektų paieška turi būti vykdoma pagal šiuos kriterijus:

Projekto pavadinimas

Užduočių paieška turi būti vykdoma pagal šiuos kriterijus:

Užduoties ID

Užduoties pavadinimas

## FR7. Projektų sąrašas gali būti eksportuojamas CSV formatu.

Pagal FR1 savybes

## FR8. Užduočių sąrašas gali būti eksportuojamas CSV formatu.

Pagal FR3 savybes

## Nefunkciniai reikalavimai

### NFR1. Sistema įgyvendinta naudojant šias technologijas:

- Java programuotojai: Java (8+) programavimo kalbą, Spring (5+) karkasą;
- Žiniatinklio programuotojai: PHP (7.4+);
- PHP programuotojai: PHP (7.4+), Laravel karkasą (8+).

## NFR2. Vartotojo sąsajai įgyvendinti naudoti atitinkamai:

- Java programuotojai: REACT;
- Žiniatinklio programuotojai: HTML, CSS, JS, BOOTSTRAP;
- PHP programuotojai: REACT.

## NFR3. Sistemos duomenys saugomi reliacinėje duomenų bazėje

## NFR4. Visi veiksmai sistemoje kaupiami įvykių žurnale

Visi veiksmai sistemoje yra kaupiami įvykių žurnale. Kiekvienas įvykis turi registruoti įvykio laiką, įvykį ir reikalingą susijusią informaciją.

## NFR5. Kartu su sistema pateikiama sistemos diegimo ir naudojimo dokumentacijos

Naudojimo dokumentacija aprašo pagrindinių sistemos funkcijų panaudojimą.  
Diegimo dokumentacija aprašo kaip įdiegti ir paleisti sistemą.

## NFR6. Visi veiksmai sistemoje turi atsakyti greičiau nei per 2 sekundes.

Reikalavimas taikomas laikantis prielaidos, kad sistemoje yra iki 1000 projektų, kurių kiekvienas turi vidutiniškai po 200 užduočių.

## Reikalavimai testavimui

1. Prieš projekto startą su projekto komanda turi būti suderinta ir laisva forma dokumentuota:

1. Kas bus testuojama.
2. Kas nebus testuojama.
3. Kokius įrankius naudojant bus ištestuoti:
  1. funkciniai reikalavimai (siūloma: testavimo atvejų valdymo įrankis - Zephyr for JIRA, klaidų registravimo ir būsenos sekimo įrankis - JIRA, testavimo ataskaitų rengimo įrankis - Zephyr for JIRA);

2. atskiri nefunkciniai reikalavimai (jeigu bus nuspręsta testuoti kuriuos nors iš jų).
4. Kokį flow komanda naudos defektų valdymui. Ar jie bus registruojami kaip atskiros JIRA užduotys, ar po susijusiu User storiu? Kokie galimi statusai, kada kuris statusas turi būti užsetinamas (pvz., kaip testuotojui atskirti, kad defektas jau yra resolved ir galima pradėti jo patvirtinimo testavimą).
2. Reikalavimai testavimo atvejams:
  1. Reikalavimai turi būti 100% padengti testavimo atvejais.
  2. Testavimo atvejai turi būti prioritetizuoti ir turėti požymius (JIRA labels), ar jie bus naudojami regresiniam testavimui ir ar jie bus automatizuojami.
3. Darbai, kurie turi būti atlikti kiekvieno sprinto metu:
  1. Sprinto pradžioje turi būti suplanuoti testavimo atvejai, kurie bus vykdomi sprinto metu. Turi būti įvertinta, ar bus reikalingas regresinis testavimas.
  2. Sprinto pabaigoje turi būti įvykdyti visi suplanuoti testavimo atvejai jų prioritetų eiliškumo tvarka.
  3. Jei reikalinga, turi būti įvykdytas regresinis ir patvirtinimo testavimas.
4. Parengta laisvos formos galutinė testavimo ataskaita, apie testavimo rezultatus.

## **Reikalavimai automatiniams testavimo atvejams**

1. Pilnas padengimas funkcionalumo atžvilgiu. Atsiskaitymo metu reikės pateikti ataskaitą, kur būtų surašyti visi testai ir kas iš jų automatizuota. Mokytojai atsitiktiniu būdu tikrins automatizuotus testus.
2. 2 suites: Smoke and Regression, turi būti galimybė juos atskirai paleisti iš command line.
3. Įgyvendintas Page Object Pattern (UI elementai ir su jais susiję metodai iškelti į Page Objects klases, BaseTest klasė su atitinkamais metodais, jei reikia, sukurtos Utils klasės).

4. Tvarkinga struktūra, testų pavadinimai turi atspindėti testuojamą sritį, turi būti suskirstyti į paketus (Package'us) pagal sritį.
5. Turi būti naudojama viena biblioteka: TestNG arba JUnit.
6. Įvedimo duomenys turi būti nuskaitomi iš failo, testai turi būti paleidžiami su keliais duomenų rinkiniais.
7. Testų išeities failai turi būti saugomi Git saugykloje šalia sistemos išeities kodo pagal komandos nustatytą projekto struktūrą.