**Vilniaus miesto vaikų darželių informacinės sistemos atnaujinimo darbų**

**Testavimo Strategija**

**2022 m.**

**Parengta**

**Akademija.it užsakymu**

**Parengė**

**Bees**

**Įvadas**

Šis dokumentas apibrėžia Vilniaus miesto vaikų darželių informacinės sistemos atnaujinimo darbųstrategiją ir laipsniškų testavimo etapų metodus, būtinus produkto pristatymo priimtinumui užtikrinti *(acceptability of the delivered solution)*. Jis apima visas produkto fazes ir išleidimus (*releases*). Testavimo ir priėmimo (*acceptance*) procesai bus grindžiami esamomis Vilniaus miesto vaikų darželių informacinės sistemos gairėmis. Šie procesai turės būti patobulinti, siekiant užtikrinti jų priėmimo (*acceptance*) kriterijus, kurie aprašomi šiame dokumente.

1. **Testavimo apimtis ir apžvalga (*Scope and overview*)**
   1. **Produkto apžvalga:**

Vilniaus miesto vaikų darželių informacinė sistema sukurta pakeisti rankinį patekimo į darželį prašymų priėmimą automatizuotu. Sistemos tikslas yra apdoroti duomenis, priskirti vaikus darželiams. Po sistemos sukūrimo visas vaikų priėmimo į darželius procesas tapo sklandesnis ir greitesnis. Šiuo metu yra suplanuoti sistemos atnaujinimo darbai, kurių metu sistema bus papildyta funkcijomis tokiomis kaip: prašymų dėl kompensacijų teikimas, atrankos atnaujinimas, saugumo užtikrinimo atnaujinimo darbai.

* 1. **Testavimo strategijos apimtis:**

Ši testavimo strategija siekia nustatyti:

* Vienas po kito einančius testavimo tipus, kurie turi būti atliekami per visą kūrimo ir eksploatavimo veiklos gyvavimo ciklą;
* Numatyti testavimo eigą ir pakeitimus, kurie vyks ir sistemos kūrimo metu;
* Kiekvieno testavimo tipo apimtį;
* Kokiu būdu turi būti pasiekti bendri lūkesčiai ir standartai visų tipų testavimui;
* Kokie yra reikalingi aukšto lygio techniniai reikalavimai;
* Reikalingas pagrindines testavimo ir kokybės užtikrinimo procedūras.
  1. **Testavimo tikslas:**

Užtikrinti programinės įrangos atitikimą funkciniams ir nefunkciniams reikalavimams.

* 1. **Testavimo aplinka:**

Vilniaus miesto vaikų darželių informacinė sistema bus pasiekiama VTMC serveryje, adresu: <https://bees.akademijait.vtmc.lt/darzelis/login>. Testavimas pradedamas tik tada, kai programuotojų kodas yra perkeliamas į Master šaką GIT saugykloje, o iš ten nukreipiamas aukščiau nurodytu adresu VTMC serveryje.

* 1. **Testavimo eiga:**

Siekiant užtikrinti, kad pirminė programos versija veikia sklandžiai ir neturi kritinių defektų, pirmiausia bus atlikti su pirmine sistemos versija gauti automatiniai testai.

Naujų funkcionalumų testavimas bus pradėtas atlikus pradinės sistemos versijos migraciją į einamuoju metu naujausias ir stabiliausias Spring Boot bei Bootstrap versijas.

Kiekvieno sprinto eigoje bus atliekami rankiniai bei automatiniai testai, patikrinantys naujai įdiegto funkcionalumo atitikimą funkciniams reikalavimams (eiliškumas eigoje gali kisti):

* Patikrinama, ar registruojant naujus vaikus sistemoje, užtenka įvesti egzistuojantį vaiko asmens kodą bei pavardę, kad kiti reikalingi duomenys apie vaiką (vardas bei gimimo data) būtų sugeneruojami automatiškai iš prijungtos išorinės sistemos.
* Patikrinama, ar prisijungęs vaiko atstovas gali pateikti prašymą kompensacijai gauti (jei vaikas lanko privatų darželį), ir ar prisijungęs švietimo specialistas pateiktus prašymus gali matyti.
* Patikrinama, ar registruojant vaiką į darželį atsirado galimybė pasirinkti papildomą atrankos kriterijų („Vaiko vieno iš tėvų deklaruojamoji gyvenamoji vieta Vilniaus savivaldybėje yra ne mažiau nei 2 metai“).
* Patikrinama, ar vaikui gavus vietą darželyje, sugeneruojamas automatinis pranešimas, kurį peržiūrėjus būtų sugeneruojama oficiali sutartis, skirta pasirašymui (pdf).
* Patikrinama, ar švietimo specialistas gali matyti įkeltas pažymas bei jas atsisiųsti.
* Patikrinama, ar vaiko atstovas bei švietimo specialistas gali matyti visas ugdymo įstaigas žemėlapyje.
* Patikrinamas sistemos slaptažodžių saugumas.

Taip pat testavimo metu bus tikrinama, ar programa atitinka nefunkcinius reikalavimus:

* Sistema veikia pakankamai greitai (į bet kurią užklausą yra atsakoma greičiau nei per 2s);
* Sistema sklandžiai veikia naudojant bet kurią iš trijų naršyklių: Google Chrome, Mozilla FireFox, Microsoft Edge (pasirenkant naujausias kiekvienos naršyklės versijas).
* Užtikrinama, kad pagrindinė sistemoje naudojama kalba – lietuvių.
* Prieš perduodant sistemą klientui, patikrinama, ar sistema sklandžiai veikia ne tik Windows, bet ir Linux operacinės sistemos aplinkoje.

Nebus testuojamas pritaikymas neįgaliesiems, pritaikymas mažesnės rezoliucijos nei 1024px mobiliems įrenginiams, našumas (stress / load tests).

1. **Testavimo metodai** 
   1. **Rankinio testavimo metodai (atlieka Testuotojai):**

**Pradžios *(entry)* kriterijai:** testavimo aplinkos prieinamumas, turimi tinkami bandymams duomenys, reikalavimai yra aiškiai apibrėžti ir patvirtinti.

1. Testavimo atvejų aprašymas:
2. Testavimo atvejai rašomi pagal atitikimo kriterijus;
3. Tik prasidėjus Sprintui, kiekvienas *user story* turi savo susijusius testavimo atvejus ir testavimo įvykdymus, kurie yra susieti su juo.
4. Testavimo atvejai yra dviejų tipų – rankiniai arba automatiniai;
5. Kiekvienas Sprintas turi savo testavimo planą (testų vykdymai, testai ir jų statusai) siekiant sekti testavimo eigą.
6. Naujų funkcijų rankinis testavimas:
7. Vartotojo sąsajos (UI) testavimas;
8. API testavimas pasitelkiant front-end;
9. Testavimas pasitelkus BrowserStack;
10. Visi testavimo atvejai turi statusus: “PASSED”, “FAILED” or “SKIPPED”;
11. Visi testavimo atvejai turi testavimo duomenis ir aktualius testų rezultatus;
12. Produkto Vadovo (PO) peržiūra:
13. Kai „ticket“ yra paruoštas PO peržiūrai – „ticket“ yra perkeliamas į PO peržiūros stulpelį;
14. Visi defektai, kurie nebuvo pašalinti iš karto, registruojami kaip atskiri „ticket“ (patvirtinus PO);
15. Jei „ticket“ patvirtinamas PO – prasideda diegimo procesas (žr. 4 skyrių).

**Išėjimo/pabaigos (*exit*) kriterijai:**

1. Atliktas regresinis testavimas.
2. Įvykdyti visi suplanuoti testavimo atvejai ir patenkinti kriterijai:
   * + Ne mažiau nei 95% aukščiausio (highest) prioriteto testavimo atvejų sėkmingi;
     + Ne mažiau nei 90% aukšto (high) prioriteto testavimo atvejų sėkmingi;
     + Ne mažiau nei 80% vidutinio (medium) prioriteto testavimo atvejų sėkmingi;
     + Parengta galutinė testavimo ataskaita.
   1. **Automatinio testavimo metodai (atlieka Testuotojai/Programuotojai):**
3. UNIT/Integraciniai testai padengs bent 60% programuotojų kodo (programuotojai):

**Pradžios (*entry*) kriterijai:** verslo reikalavimai pristatyti ir yra aiškūs.

**Išėjimo/pabaigos (*exit*) kriterijai:** sėkmingai įvykdyti unit/integraciniai testai; unit/integraciniai testai padengia verslo reikalavimus ir kraštutinius atvejus.

1. API testavimas (Testuotojai/Programuotojai):

**Pradžios (*entry*) kriterijai:** verslo reikalavimai pristatyti ir yra aiškūs.

**Išėjimo/pabaigos (*exit*) kriterijai:** sėkmingai įvykdyti API testai; API testai padengia verslo reikalavimus ir kraštutinius atvejus.

1. E2E/priėmimo (acceptance) testai (Testuotojai/Programuotojai):

**Pradžios (*entry*) kriterijai:** buvo įvykdytiverslo reikalavimai.

**Išėjimo/pabaigos (*exit*) kriterijai:** apžvelgtas ir testais padengtas esminis sistemos procesas; nėra paliktos jokios kritinės klaidos.

Sistemos testavimui bus naudojami sistemos kūrimo metu paruošti prisijungimai bei slaptažodžiai:

* prisijungimo vardas: admin@admin.lt, slaptažodis: [admin@admin.lt](mailto:admin@admin.lt) – administratoriui;
* prisijungimo vardas: manager@manager.lt, slaptažodis: [manager@manager.lt](mailto:manager@manager.lt) – švietimo atstovui;
* prisijungimo vardas: user@user.lt, slaptažodis: [user@user.lt](mailto:user@user.lt) – vaiko motinai / tėvui / globėjai(-ui).

Jei testavimo atvejams patikrinti reikės kitų vartotojų duomenų, jie bus sukurti testavimo metu.

1. **Testavimo klaidos**

**3.1.Testavimo klaidų valdymas**

Testavimo procesas valdomas JIRA aplinkoje. Testavimas pradedamas, kai užduotis yra perkeliama iš „in code review“ skilties į „ready for QA“ skiltį.

Testavimo atvejai aprašomi naudojant Zephyr for JIRA. Testavimo atvejai susiejami su konkrečiais *user stories*. Jei testo metų defektų nėra randama, jie pažymimi kaip sėkmingi. Radus defektą, jis yra užregistruojamas ir perduodamas programuotojams, kad būtų pašalintas (būsena pakeičiama į „in progress“). Kai defektas yra pašalinamas, užduoties būsena yra perkeliama į „done“.

**3.2. Testavimo ataskaitų rengimas**

Rankinio ir automatinio testavimo procesų ataskaita bus prieinama JIRA QA dashboard aplinkoje. Planuojama, kad ataskaitoje bus aprašyti rankinio bei automatinio testavimo rezultatai, detalizuota, kiek buvo atlikta testavimo atvejų ir kiek iš jų buvo sėkmingi.

1. **Testavimo įrankiai**

Testavimo procesas valdomas JIRA aplinkoje.

Pritaikymas skirtingoms naršyklėms testuojamas pasitelkiant Google Lighthouse (Google Chrome Dev Tools) ir Google Chrome Dev Tools Network.

Sistemos saugumas tikrinamas naudojantis Postman.

Automatiniai UI testai kuriami IntelliJ aplinkoje, naudojant JAVA, Selenium WebDriver ir TestNG. Kodas saugomas GIT saugykloje kartu su programuotojų paruoštu sistemos kodu.