

Vaikų darželių informacinė sistema

Reikalavimų specifikacija (2021 v1.0)

Įvadas

Šis dokumentas aprašo darželių informacinės sistemos poreikį miesto savivaldybei ir reikalavimus. Informacinė sistema leistų operatyviau priimti prašymus lankyti ugdymo įstaigą, atlikti eilės skaičiavimus, pateikti statistinius duomenis.

IS poreikio analizė

Vilniaus miesto seniūnijose yra virš 140 vaikų darželių. Preliminariai ikimokyklinio ugdymo įstaigas lanko apie 30 tūkst. vaikų. Jau dabar aišku, kad visi vaikai neturės galimybės gauti vietą darželyje – likusieji gaus vietos numerį laukti eilėje. Miesto savivaldybė kasmet fiziškai priima prašymus, juos apdoroja, priskiria vaikus darželiams, kas reikalauja daug resursų ir laiko. Neskaitant patiriamų išlaidų resursams, eilių formavimas užtrunka dar ilgiau, todėl šeimos, turinčios vietas darželiuose, yra informuojamos pavėluotai, užtrunka ir tampa sudėtingas informacijos valdymas.

Reikalingas sprendimas

Reikalinga vaikų registracijos į darželius informacinė sistema Vilniaus miesto savivaldybei, kad būtų galima automatizuotai priimti ir valdyti prašymus, atrinkti prašymus pagal prioritetus, sudaryti eiles darželiams, o tėvams būtų suteikta galimybė matyti ir valdyti savo vaiko informaciją. Tokia sistema suteiktų tikslią, patikimą, skaidrią informaciją.

Naudotojai ir rolės

Administratorius	<i>Miesto savivaldybės specialistas, atsakingas už sistemos saugią konfigūraciją ir palaikymą</i>
Švietimo specialistas	<i>Informacinės sistemos tvarkytojas, atsakingas už centralizuoto vaikų priėmimo į švietimo įstaigas duomenų tvarkymą, tvarkantis įstaigų ir prašymų duomenis.</i>
Vaiko atstovas	<i>Tėvai arba globėjai, turintys teisę naudotis informacine sistema.</i>

Funkciniai reikalavimai

1. Prisijungti prie sistemos tėvams:

Naudojant administratoriaus sukurtus prisijungimo duomenis. Administratorius turi galėti kurti vaiko atstovus ir priskirti jiems rolę. Pirminis slaptažodis - toks pats kaip naudotojo prisijungimo vardas. Administratorius gali nustatyti slaptažodį į numatytąjį (pirminį). Naudotojas prisijungęs gali pasikeisti slaptažodį į bet kokią norimą slaptažodį. Apribojimai keičiamam slaptažodžiui: slaptažodis turi būti ne mažiau 8 simbolių ilgio, turi būti bent viena didžioji ir bent viena mažoji raidė, ir bent vienas skaičius.

2. Užregistruoti prašymą lankyti ikimokyklinio švietimo įstaigą.

Prašymas yra registruojamas prisijungus prie IS nurodant tėvų/globėjų ir vaiko asmeninius duomenis ir pasirenkant daugiausiai 5 įstaigas pagal šeimos pageidaujamą prioritetą. Taip pat pasirenkami prioriteto kriterijai, turintys įtakos pirmumui.

3. Valdyti savo (tėvų) duomenis sistemoje.

Tėvai/globėjai, prisijungę prie sistemos gali atnaujinti savo kontaktinius duomenis, sukurti ar atšaukti naujus prašymus lankyti įstaigas. Naujas prašymas anuliuoja prieš tai esantį aktyvų prašymą.

4. Sukurti paskyras vidiniams sistemos naudotojams

Administratorius turi galėti kurti papildomas paskyras, reikalingas savivaldybės švietimo specialistams.

5. Sukurti laukiančiųjų eilę

Sistemoje turi būti galimybė sukurti eilę į kurią bus registruojami visi prašymai (pvz. 2021 pirmasis kvietimas į darželius). Turi būti galimybė sustabdyti registracijų priėmimą.

6. Sistema surūšiuoja registracijų sąrašą ir pasiūlo patvirtinimui

Turi būti galimybė inicijuoti registracijų rūšiavimą pagal savivaldybės kriterijus.

Surūšiuoti duomenys turi būti išvesti į vartotojo sąrašą ir pateikti patvirtinimui.

Patvirtinus sąrašą jis turi būti išsaugomas. Turi būti galimybė užrakinti ir atrakinti sąrašo redagavimą (administratoriui).

7. Koreguoti priskyrimo darželiui duomenis

Švietimo specialistas turi turėti galimybę koreguoti surūšiuotus duomenis, kol jie dar nėra patvirtinti.

8. Sudaryti laukiančių vaikų eilę pagal prioritetus.

Nepriskirti grupėms vaikai yra surikiuojami (pagal tuos pačius kriterijus) į laukiančiųjų eilę su eilės numeriu.

9. Pateikti statistinius duomenis (tėvams ir švietimo specialistams):

- *Peržiūrėti visų užregistruotų, priimtų ir eilėje laukiančių vaikų skaičius;*
- *Peržiūrėti pagal prioritetus darželių pavadinimus (1, 2, 3, 4, 5 prioritetas);*
- *Peržiūrėti eilėje laukančius prašymus pagal prioritetus (įstaigų sąrašas pagal 1, 2, 3, 4, 5 prioritetą).*

10. Archyvuoti asmens duomenis.

Pagal tėvų reikalavimą pašalinti jo duomenis iš sistemos (pateiktą kitais kanalais), administratorius (prieš panaikinimo archyvavimą) turi turėti galimybę suarchyvuoti visus pareiškėjo duomenis, kad galima būtų juos pateikti persiuntimui. Turi būti galimybė peržiūrėti archyvą.

Nefunkciniai reikalavimai

1. Programai kurti naudojamos Java, Spring, React.js technologijos, reliacinė duomenų bazė.
2. Į bet kurią užklausą sistema turi atsakyti greičiau nei per 2 sekundes (web puslapio užkrovimas).
3. Jeigu sistemoje bus naudojami slaptažodžiai - sistemos slaptažodžiai negali būti saugomi atviru tekstu.

Duomenų bazėje saugomi slaptažodžiai negali būti saugomi atviru tekstu ar šifruoti. Slaptažodžiai turi būti paslėpti taikant SHA-2 šeimos ar ne mažesnio atitinkamo saugumo maišos funkciją.

4. Visi sistemos veiksmai kaupiami įvykių žurnale, turi būti realizuota galimybė peržiūrėti įvykius naudotojo sąsajoje (UI).

Visi veiksmai sistemoje kaupiami įvykių žurnale, kartu su juos atliekančių vartotojų identifikaciniais duomenimis bei kitu, konkrečiam įvykiui reikiamu kontekstu.

Įvykių žurnale registruojama:

- visos vartotojų matomos ir nematomos klaidos
- vartotojų prisijungimai
- nesėkmingi prisijungimo bandymai
- visos būseną keičiančios operacijos (įskaitant vartotojo duomenis, kurie jas atlieka)

5. Kartu su sistema pateikiama diegimo, panaudojimo dokumentacija.

Pateikiama išsami pagrindinių sistemos funkcijų dokumentacija.

6. Sistema turi būti suderinama su Linux operacine sistema.
7. Programuojamas kodas turi būti padengtas unit testais bent 60 %.
8. Sistemos pridavimo metu, sistema turi teisingai veikti su Google Chrome, Mozilla FireFox ir Microsoft Edge naršyklių to meto naujausiomis versijomis.
9. Pagrindinė sistemoje naudojama kalba - lietuvių.

Vaikų priėmimo tvarkos prioritetai (eilės sudarymas):

#	Prioritetas (pagal pirmumą)
1	Vaiko deklaruojama gyvenamoji vieta yra miesto savivaldybė;
2	Įvaikinti vaikai;
3	Vaikams iš šeimų, auginančių (globojančių) tris ir daugiau vaikų, kol jie mokosi pagal bendrojo ugdymo programas;
4	Vaikams, kurių brolis ar sesuo lanko tą pačią įstaigą (pirmumo teisė įsigalioja iškart, prioritetas taikomas tik toje pačioje įstaigoje, kurią tuo metu lanko brolis ar sesuo);
5	Vaikams, kurių vienas iš tėvų (globėjų) mokosi bendrojo ugdymo mokykloje;
6	Vaikams, kurių vienas iš tėvų (globėjų) turi ne daugiau kaip 40 procentų darbingumo lygio;

Papildoma informacija eilės sudarymui:

- Registracijos data ir laikas **nėra** prioriteto kriterijus.
- Jei prašyme pirmu numeriu nurodytoje įstaigoje nėra laisvų vietų, vieta skiriama antru numeriu pažymėtoje įstaigoje, jei joje yra laisvų vietų, ir t. t. Jeigu visose prašyme pažymėtose įstaigose nėra laisvų vietų, prašymas lieka laukiančiųjų eilėje.
- Patikrinus IS duomenis apie pirmenybę skirti vaikui vietą įstaigoje ir nustatčius vienodą pirmenybių skaičių, vieta eilėje skiriama vyresnio pagal gimimo datą amžiaus vaikams, o, sutapus vaikų gimimo datai, eilė sudaroma abėcėlės tvarka pagal vaiko pavardę.
- Nesant pirmenybių, vieta eilėje skiriama vyresnio amžiaus vaikams.

- Į laisvas vietas jau sudarytose grupėse po pagrindinės metų atrankos priimama nuolat.

Reikalavimai testavimui

1. Prieš projekto startą su projekto komanda turi būti suderinta ir laisva forma dokumentuota:
 1. Kas bus testuojama.
 2. Kas nebus testuojama (pvz., gal bus netestuojamų User storių ar funkcinių reikalavimų; galbūt nebus testuojami kurie nors iš nefunkcinių reikalavimų (saugumas, pritaikymas neįgaliesiems, našumas, naršyklių/mobilių įrenginių suderinamumas, naudotojo sąsaja ar UX'as)).
 3. Kokius įrankius naudojant bus ištestuoti:
 1. funkciniai reikalavimai (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. Aplinka(-os), kurioje(-ose) bus testuojami naujausi kodo pakeitimai. Aplinka, kurioje bus vykdomas regresinis sistemos testavimas prieš diegimą į produkcinę aplinką (jeigu bus nuspręsta turėti tokią papildomą pre-prod / staging aplinką).
 5. Kas ir kaip generuos testavimui reikalingus, panašius į produkinius duomenis. Ar yra toks poreikis iš viso, gal pakaks pačių testuotojų sugeneruotų duomenų testavimo metu? Ar yra poreikis reguliariam testavimo duomenų atnaujinimui į pradinę būseną?
 6. 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, ar jie bus naudojami regresiniam testavimui ir ar jie bus automatizuojami.
3. Pirmo sprinto testavimo atvejai turi būti patvirtinti QA mentorių.
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 ir patvirtinimo 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. Darbai, kurie turi būti atlikti prieš diegimą į produkcinę aplinką:
 1. Atliktas regresinis testavimas
 2. Įvykdyti visi suplanuoti testavimo atvejai ir patenkinti kriterijai:
 1. $\geq 95\%$ aukščiausio (highest) prioriteto testavimo atvejų sėkmingi
 2. $\geq 90\%$ aukšto (high) prioriteto testavimo atvejų sėkmingi
 3. $\geq 80\%$ vidutinio (medium) prioriteto testavimo atvejų sėkmingi
 4. Liko neišspręstas ne daugiau kaip 1 kritinis (highest priority) defektas; defektui turi būti įvardintas sprendimas, patvirtintas PO
 5. Liko neišspręsti ne daugiau kaip 2 reikšmingi (high priority) defektai
5. Parengta laisvos formos galutinė testavimo ataskaita, įrodanti 4.2 punkte išvardintus teiginius.

Reikalavimai automatiniam 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, naudojamas PageFactory, BaseTest klasė su atitinkamais metodais, jei reikia, sukurtos Utils klasės).
4. Tvarkinga struktūra, testų pavadinimai turi atspindėti testuojamą sritį, testuose turi būti surašytas scenarijus JavaDoc formatu, 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.