Latvijas Republikas izglītības un zinātnes ministrija Daugavpils Tehnoloģiju un tūrisma tehnikums

PROJEKTA DARBS

Programmēšanas tehniķis

specialitāte

1. projekta darbs

temats

Testēšanas sistēmas izveidošana

PASKAIDROJOŠAIS RAKSTS

**DTTT.P.D.0009.009PR**

Profesijas kods 33 481 031 Programmēšanas nodaļa PR-21 grupa

|  |  |  |
| --- | --- | --- |
| Izveidoja | paraksts | R.Giruckis  V.,Uzvārds |
| Pārbaudīja | paraksts | I. Dortiņa  V.,Uzvārds |

2024./2025. m.g.

**Saturs**

[Ievads 3](#_Toc193962522)

[1. Uzdevuma nostādne 4](#_Toc193962523)

[1.1. Sistēmas modelis 4](#_Toc193962524)

[1.1.1. Priekšmetiskās jomas informācijas modelis 4](#_Toc193962525)

[1.1.2. Sistēmas apkārtnes shēma 5](#_Toc193962526)

[1.1.3. Viedokļu analīze 5](#_Toc193962527)

[1.1.4. Datu modelēšana 7](#_Toc193962528)

[1.1.5. Datu vārdnīca 8](#_Toc193962529)

[1.2. Sistēmas evolūcija 10](#_Toc193962530)

[1.3. Sistēmas funkcionālās prasības 10](#_Toc193962531)

[1.4. Sistēmas nefunkcionālās prasības 14](#_Toc193962532)

[2. Priekšmetiskās jomas klašu sistēmas izstrāde 24](#_Toc193962533)

[2.1. Klašu diagramma 25](#_Toc193962534)

[2.2. Klašu realizēšana Java valodā 26](#_Toc193962535)

[3. Testa programmas izstrāde 29](#_Toc193962536)

[3.1. Testēšanas metodikas 29](#_Toc193962537)

[3.2. Testēšanas programmas struktūra 30](#_Toc193962538)

[3.3. Testēšanas rezultāti 39](#_Toc193962539)

[4. Lietotāja rokasgrāmata 41](#_Toc193962540)

[4.1. Lietotāja instrukcija 41](#_Toc193962541)

[4.2. Palīdzības sistēma (Help) 43](#_Toc193962542)

[Secinājumi 45](#_Toc193962544)

[Izmantotās literatūras (informācijas avotu) saraksts 46](#_Toc193962545)

[Pielikums A. Klašu sistēma 47](#_Toc193962546)

[Pielikums B. testa programmas kods 53](#_Toc193962547)

Ievads

Projekta darbā ir aprakstīta testēšanas programma, kura izpilda sekojošas funkcijas:

* lietotāju reģistrēšana
* datu ievade
* datu saglabāšana
* datu rediģēšana
* testa pildīšana
* rezultātu aprēķināšana

Sistēma nodrošina audzēkņa testēšanu par noteiktu tēmu/vielu, tas ir domāts, lai skolotājs varētu viegli izlikt atzīmi skolēnam digitālā veidā attālināti vai klātienē. Tāpat šī programma novērš cilvēcisko faktoru, piemēram, noapaļošanu noteiktā virzienā.

Programmatūra ir izstrādāta tā, lai skolotājam būtu viegli pārbaudīt skolēnus. Skolotājam ir jāizvēlas tests, kas tiks dots skolēnam, un skolēnam ir tikai jāaizpilda tests. Programmatūra ir paredzēta skolēniem, kuri slimības vai citu iemeslu dēļ ir kavējuši mācības, un, lai nesabojātu savu vērtējumu, viņi var kārtot testu vai arī, ja skolotājs to atļauj, kārtot treniņ tēstu. Programma parādīs skolēna lietotājvārdu, testu, un atzīmi, un skolotājam sistēmā būs tikai jāievada rezultāti.

Lai ienakt ka administrators:

Lietotājvārds: admin

Parole:123

# Uzdevuma nostādne

## Sistēmas modelis

### Priekšmetiskās jomas informācijas modelis

Apskatāmā sistēmā tika izdalīti objekti: Datu bāze (Fails ar atbildēm), Tests, Rezultāti, Administrators / Skolotājs, Lietotājs. Šie objekti un saites starp tiem ir paradīti ([1.1. att.](#att1_1)).

Datu bāze

Skolotājs

Tests

Rezultāti

Lietotājs

Treniņš

1.1 att. Vienkāršots testēšanas sistēmas modelis

Saišu nozīme vienkāršotā nodaļas modelī:

* Skolotājs ir testa vai treniņ testa atļautajs un pārbaudītājs, kuram ir pieeja pie atbildēm, testam un lietotāja rezultātiem;
* Datu bāze satur failus ar jautājumiem un ar pariezām atbildēm;
* Lietotājs ir testa vai treniņ testa pildītājs, kas beigās var apskatīt savus rezultātus, nepieciešamības gadījumā viņš varēs sazināties ar skolotāju;
* Tests ir instruments, kas palīdzēs pārbaudīt lietotāju zināšanas un novērtēt to ar atzīmi;
* Rezultāti parāda atzīmi un procentuāli.

### Sistēmas apkārtnes shēma

Dati, kā atbildes, ko ievada audzēknis tiek saglabāti failā. Visas atbildes ir automātiski novērtētas. Saskarni ar lietotājiem nodrošinās interfeisa modulis ([1.2. att.](#att1_2)).

Lietotājs

Lietotāja saskarne

Sistēma

Datu bāze

Lietotāja dati un rezultāti

## att. Sistēmas apkārtnes modeļa piemērs

### Viedokļu analīze

Savāktie un identificētie viedokļi par sistēmu ir attēloti ar burbuļdiagrammas palīdzību ([1.3. att.](#att1_3)), tas faktiski ir viedokļu kopums, kur katra viedokļa nosaukums ir ierakstīts atsevišķa elipsē.

Nefunkcionalie viedokli

Funkcionalie viedokli

Viedokli par lietotaju

Datu viedokļi

1.3. att. Viedokļu burbuļu diagramma

Otrais etaps viedokļu analīzē ir viedokļu klasifikācija, kad viedokļi ir sagrupēti grupās ([1.4. att.](#att1_4)).

Testēšanas sistēma (V0)

Datu bāze (V12)

Lietotājs (V11)

Tests (V13)

Lietotāju registrt (V121)

Skolotājs (V111)

Testa jautajumu registrs (V122)

Audzēknis (V112)

Audzekņa funkcijas (V132)

Skolotāja funkcijas (V131)

Testa pildīšana (V1321)

Testa atļaušana (V1311)

Testa rezultātu saņemšana (V1323)

Audzekņa testa atzime skatīšana. (V1312)

Treniņā

pildīšana (V1322)

Treniņa rezultātu saņemšana (V1324)

Vai tests ir pieejams(V123)

Admins (V113)

Treniņa atļaušana (V1313)

1.4. att. Viedokļu hierarhiskā struktūra

### Datu modelēšana

Datu modelēšanai ir tiek izmantots ER modelis. Testēšanas sistēmas ER modelis ([1.5. att.](#att1_5)) ietver datu entītiju kopumu un relāciju kopumu starp datu entītijām, pie kam ar šo modeli var attēlot dažāda tipa relācijas starp datu elementiem.

Audzēknis

Pilda

Tests

Skolotājs

Atļaut

Jautājums

Satur

Tests

Admins

1.5. att. Vienkāršots testēšanas sistēmas ER modelis

### 

### Datu vārdnīca

|  |  |  |
| --- | --- | --- |
| **Datu tips** | **Nosaukums** | **Apraksts** |
| **String** | **text1** | Vārds |
| **String** | **text2** | Uzvārds |
| **String** | **text3** | Lietotājvārds, kuru ievada lietotājs |
| **String** | **text4** | Parole, kuru ievada lietotājs |
| **String** | **text5** | Parole atkārtoti pārbaudei |
| **String** | **role** | Lietotāja loma: "student" ,"skolotajs" vai "admin" |
| **boolean** | **found** | Pārbauda, vai fails vai atbildes eksistē un ir derīgas |
| **int** | **correctAnswersCount** | Pareizi atbildēto jautājumu skaits |
| **int** | **totalQuestions** | Kopējais jautājumu skaits (20) |
| **double** | **grade** | Atzīmes aprēķins pēc pareizo atbilžu skaita |
| **int** | **roundedGrade** | Atzīme kā vesels skaitlis pēc noapaļošanas |
| **double** | **percentage** | Aprēķinātais pareizo atbilžu procentuālais daudzums |
| **List<Integer>** | **answerResults** | Saraksts ar 1/0 vērtībām katram jautājumam (pareizi/nepareizi) |

## Sistēmas evolūcija

Testēšanas sistēma ir neatkarīga un pašpietiekama programma. Produktam nav paredzētas vairākas sastāvdaļas.

Programmai ir trīs ārējās saskarnes, kuras var apskatīt 1.4. nodaļā – viena paredzēta reģistrēšanai sistēmā, otrā – ielogošanai sistēmā un trešā – testa pildīšanai. Grafisko lietotāja saskarni var pilnveidot, papildinot to ar dažādām krasu un noformējuma shēmām.

Testēšanas sistēmā paredzēti divu lietotāju veidi: administrators un lietotājs, kurš pilda testu. Nākotnē plānots pievienot vēl vienu lietotāju grupu ar funkciju veidot jaunus testus un pārbaudīt izpildītos testus, līdz ar to lietotājiem būs divas lomas: audzēknis, kurš pilda testu un skolotājs, kurš veido un pārbauda testus. Administratoram plānots realizēt funkciju mainīt lietotāja vārdu, uzvārdu, lietotājvārdu vai paroli, ka arī pievienot un dzēst lietotājus un mainīt vīnu lomas.

Programma satur vienu testu, kurā ir 10 jautājumi, tomēr nākotnē sistēmu var pilnveidot, pievienojot jaunas testa tēmas, kļūdu uzrādīšanu un kļūdu skaidrojumu pēc testa izpildes, lietotāju reitingu, lietotāju datu izvadi lietotāja izvēlnē vai administratora izvēlnē.

Sistēmas dažiem datu tipiem tiks izmantoti dati, kuri tiek saglabāti teksta failā, nākotnē programmu var pilnveidot darbam ar datu bāzi (datu bāzes savienojums, testa jautājumi un atbildes uz tiem u.c.)

## Sistēmas funkcionālās prasības

1. **Funkcija “Reģistrēties sistēmā”**

***ID: T1***

**Ievads**: Ļauj ievadīt nepieciešamus datus, lai izveidotu savu kontu. Visi ievadīti dati tiek saglabāti failā (datu bāzē).

Ievade:

1. Lietotāja vārds
2. Lietotāja uzvārds
3. Lietotāja lietotājvārds
4. Lietotāja parole
5. Ievadītās paroles apstiprināšana.

**Apstrāde**:

1. Lietotāja vārda saglabāšana failā.
2. Lietotāja uzvārda saglabāšana failā.
3. Lietotāja lietotājvārda saglabāšana failā.
4. Lietotāja paroles saglabāšana failā.

Ja ievadītā parole ir nepareiza, tad tiek atvērts kļūdas dialoglodziņš.

**Izvade**: Dialoglodziņš, kurā uzrakstīts “Reģistrēšana ir veiksmīgi pabeigta!”.

1. **Funkcija “Ielogoties sistēmā”**

***ID: T2***

**Ievads**: Ļauj ievadīt nepieciešamus datus, lai ieiet sava kontā.

**Ievade**:

1. Lietotāja lietotājvārds
2. Lietotāja parole

**Apstrāde**:

1. Lietotāja lietotājvārda pārbaude un atrašana DB.
2. Lietotāja paroles pārbaude un atrašana DB.

Ja ievadītā parole vai/un lietotājvārds ir nepareiza, tad tiek atvērts kļūdas dialoglodziņš.

**Izvade**: Dialoglodziņš, kurā uzrakstīts “Ieeja ir veiksmīgi pabeigta!”.

1. **Funkcija “Sākt testu”**

***ID: T3***

**Ievads**: Ļauj administratoram sākt testu.

**Ievade**: Administrators noklikšķina uz pogu “Sākt testu”. Pēc tam sistēma uzdod jautājumu: "Vai Jūs vēlaties sākt testu?" Administratoram ir divi varianti atbildēm "Jā" vai "Nē".

**Apstrāde**: Sistēma pārbauda atbildi uz uzdoto jautājumu un pēc tam sāk testu, ja uz uzdoto jautājumu atbilde "Jā". Tad tests ir pieejams lietotājiem to veikšanai.

**Izvade**: Administratora ekrānā tiek parādīts dialoglodziņš ar tekstu "Tests ir atļauts!"

1. **Funkcija “Pildīt testu”**

**I*D: T4***

**Ievads**: Ļauj lietotājam sākt testa izpildi.

**Ievade**: Lietotājs noklikšķina uz pogu “Pildīt testu”.

**Apstrāde**: Ja tests ir pieejams izpildei, tad pēc testa izvēles tiek parādīts dialoglodziņš ar tekstu "Tests ir pieejams! Vai vēlaties izpildīt testu?". Lietotājam ir divi varianti atbildēm "Jā" vai "Nē". Ja tests nav pieejams izpildei, tad pēc testa izvēles tiek parādīts dialoglodziņš ar tekstu "Tests nav pieejams!"

**Izvade**: Lietotājā ekrānā tiek atvērts logs ar testu.

1. Atvērts logs ar Testu
2. **Funkcija “Atbildēt uz testa jautājumu”**

***ID: T5***

**Ievads**: Ļauj ievadīt atbildi uz testa jautājumus.

**Ievade**: Lietotājs izvēlas 1 no piedāvātajām atbildēm uz uzdoto jautājumu.

**Apstrāde**: Sistēma pārbauda atbildi uz jautājumu. Pēc katras pareizās atbildes sistēma pievieno 1.

**Izvade**: Pēc izvēlētās atbildes lietotājs noklikšķina uz pogas "nākamais jautajums", pēc kura tiek parādīts nākamā izvēlne ar testa jautājumiem.

1. **Funkcija “Parādīt rezultātu”**

***ID: T6***

**Ievads**: Ļauj uzzināt nokārtotā testa rezultātus.

**Ievade**: Lietotājs noklikšķina uz pogas "Parādīt rezultātu".

**Apstrāde**: Sistēma saskaita pareizo atbilžu skaitu, nepareizo atbilžu skaitu un neatbildēto jautājuma skaitu, tad aprēķina pareizo atbilžu procentuālo vērtību, pēc tam nolasa vērtējumu par 10 balles skalu un atrod komentāru par saņemto atzīmi.

**Izvade**: Tiek atvērts dialoglodziņš, kurā tiek rakstīts šāds teksts:

1. Neatbildēto jautājuma skaits.
2. Pareizo atbilžu skaits.
3. Nepareizo atbilžu skaits.
4. Pareizo atbilžu procentuālā daļa.
5. Atzīme.
6. Sistēmas komentārs par saņemto atzīmi.
7. **Funkcija “Salidzini Rezultatie ar Videjie”**

***ID: T7*** **Ievads**: Ļauj salīdzināt skolēna rezultātus ar vidējiem klase

**Ievade**: Lietotājs noklikšķina uz pogas " Salīdzini rezultatie ar vidējie klase".

**Apstrāde**: Sistēma nosaka vidējo atzīme.

**Izvade**: Tiek atvērts dialoglodziņš, kurā tiek rakstīts šāds teksts:

1. Vidēja atzīme/ procenti.
2. Skolēnu atzīme/procenti.
3. **Funkcija “Iespēja apskatīt pareizās atbildes”**

***ID: T8*** **Ievads**: Iespēja redzēt pareizās atbildes tikai pēc tam, kad skolotājs ir devis atļauju to darīt.

**Ievade**:Skolotajs noklikšķina uz pogas pirms testa uzsākšana "Atļaut skatit pareizās atbildes" un pec tam Lietotājs noklikšķina uz pogas " Skatīt pareizās atbildes" pec testa nobeigums.

**Apstrāde**: Sistēmā jau ir pareizās un skolēna atbildes uz jautājumiem.

**Izvade**: Tiek atvērts dialoglodziņš, kurā tiek rakstīts šāds teksts:

1. Skolēna atbildes.
2. Pareizās atbildes.
3. **Funkcija “Skolotajs var apskatīt visu skolēnu atzīmes”**

***ID: T9*** **Ievads**: Iespēja skatīt visu skolēnu darbus rezultātus var redzēt atzīmi par 1 un 2 tests.

**Ievade**: Skolotajs noklikšķina uz pogas " Apskatīt visu skolēnu atzīmes 1.Tests "

vai " Apskatīt visu skolēnu atzīmes 2.Tests ".

**Apstrāde**: Sistēma iegūst skolēnu rezultatus un procentus no txt faila..

**Izvade**: Tiek atvērts dialoglodziņš, kurā tiek rakstīts šāds teksts:

1. Skolēnus atzīmes in procentus par izvēlētais tests.
2. **Funkcija “Iespēja Trenieties”**

***ID: T10*** **Ievads**: Iespēja nokārtot nelielu testu, kas būs saistīts ar testa tēmu, bet tikai tad, ja administrators dod iespēju praktizēties.

**Ievade**: Lietotājs noklikšķina uz pogas " Treniņš".

**Apstrāde**: Administrators var izvēlēties, vai veikt vai neveikt papildu testu, lai pārbaudītu zināšanas,.Ja treniņš nav pieejams izpildei, tad pēc treniņa izvēles tiek parādīts dialoglodziņš ar tekstu "Treniņš nav pieejams!".

**Izvade**: Tiek atvērts logs, kurā kur tiks uzdots neliels tests.

1. Atvērts logs ar Treniņu
2. **Funkcija “Palīdzība”**

***ID: T11***

**Ievads:** Ļauj lietotājam lai uzzinātu, ko no viņa prasa programma

**Ievade:** Lietotājs noklikšķina uz pogas “Help”  **Apstrāde:** Sistēma atver dialoglogu ar visiem norādījumiem, lai lietotājs saprastu, kas no viņa tiek prasīts.

**Izvade:** Lietotājā ekrānā tiek parādīts dialoglodziņš ar uzrakstu kas no viņa tiek prasīts programmā.

1. Atvērts dialoglogs ar palidzibu

## Sistēmas nefunkcionālās prasības

1. Prasības produktam:
   1. Lietotāju saskarne ar sistēmu notiek latviešu valodā.
   2. Produktam nav paredzētas vairākas sastāvdaļas. Produktam ir trīs ārējās saskarnes: reģistrēšanai sistēmā, ielogošanai sistēmā, testa pildīšanai.
   3. Programma ir paredzēta vienam lietotājam, kurš saprot latviešu valodu. Lietotāja vecuma vai izglītības ierobežojumi nav paredzēti.
2. Ārējās saskarnes prasības
   1. Lietotāja saskarne: produkts paredz vienotu lietotāja saskarni.
   2. Ekrāna formāti: minimālais ekrāna izmērs ir 10” ar minimālo izšķirtspēju –800x600 punkti.
   3. Lietotāja saskarne “Lietotāja reģistrēšanas saskarne”. Saskarnes struktūra: skat. [1.6. att.](#att1_6)

Reģistreties

Uzvards:

Lietotajvards:

Parole:

Parole (atkartoti):

Vards:

Ielogoties

**reģistrēšanā**

Es esmu

* skolotājs
* skolēns

1.6. att. Lietotāja reģistrēšanas saskarne

**Saskarnes elementi:**

1. teksta lauks “Vārds”
2. teksta lauks “Uzvārds”
3. teksta lauks “Lietotājvārds”
4. teksta lauks ‘Parole”
5. teksta lauks “Parole (atkārtoti)”
6. Izvelē “Skolotājs vai skolēns”
7. poga “Reģistrēties”
8. poga “ Ielogoties”
9. poga “ Help”
10. poga “ Exit”
    1. Lietotāja saskarne “Lietotāja ielogošanas saskarne”. Saskarnes struktūra: skat. [1.7. att.](#att1_7)

Reģistreties

Lietotājvārds:

Parole:

Ielogoties

**Lietotāja pieslēgšanā**

Help

Exit

1.7. att. Lietotāja ielogošanas saskarne

**Saskarnes elementi:**

1. teksta lauks “Lietotājvārds”
2. teksta lauks ‘Parole”
3. poga “Ielogoties”
4. poga “Reģistrēties”
5. poga “ Help”
6. poga “ Exit”
   1. Lietotāja saskarne “Testa izpilde”. Saskarnes struktūra: skat. [1.8. att.](#att1_8)

a) b)

Sakt testu

Tests “Nosaukums”

Sakt treniņu

Help

Uz testa izvele

Exit

nākamais jautājums

1. Jautajums “Jautajuma teksts”

1. atbilde

2. atbilde

3. atbilde

Help

e)

1. Tests

Testu Izvēle

2. Tests

Testa “Nosaukums”

Testa “Nosaukums”

Iziet no sistēmas

Help

Exit

Help

d)

Pabeigt

Pareizas atbildes

1. ”Atbilde”
2. ”Atbilde”

…”Atbilde”

c))

Beigt

Rezultati

Procenti: X%

Atzime: A

Skatīt pareizās atbildes

Salidzini RezultatiearVidejie

Help

Exit

1.8. att. Testa izpildes saskarne: testa izvēle/sakums; b) atbilde uz jautājumu/-iem ; c) rezultāti d) pareizās atbildes

Saskarnes elementi:

a)

1. Teksts”Tests Nosaukums”
2. Poga “Sakt testu”
3. Poga “Sakt treniņu”
4. Poga “Uz testa izvele”
5. Poga “Help”
6. Poga “Exit”

b)

1. Teksts” Jautajums/Jautajuma teksts”
2. Atbilžu izvēle
3. Poga “Talak”
4. Poga “Help”

c)

1. Teksts”Rezultati”
2. Teksts”Procenti”
3. Teksts”Atzime”
4. Poga “Beigt”
5. Poga “Skatīt pareizās atbildes”
6. Poga “Salidzini Rezultatie ar Videjie”
7. Poga “Help”
8. Poga “Exit”

d)

1. Teksts” Pareizas atbildes”

2. Teksts”Atbilde”

3. Teksts” Pabeigt”

4. Poga “Help”

e)

1. Teksts” Testu Izvēle”
2. Teksts” Testa “Nosaukums” ”
3. Teksts” Testa “Nosaukums” ”
4. Poga “Help”
5. Poga “Exit”
6. Poga “Iziet no sistēmas”
7. Poga “1. Tests”
8. Poga “2. Tests”
   1. Lietotāja saskarne “Treniņa izpilde”. Saskarnes struktūra: skat. [1.9. att.](#att1_8)

nakamajs jautajums

2.Jautajums “Jautajuma teksts”

1. atbilde

2. atbilde

3. atbilde

Help

a)

b)

Beigt

Treniņa Rezultati

Procenti: X%

Atzime: A

Help

1.9. att. Testa izpildes saskarne: treniņa sakums; b) atbilde uz jautājumu/-iem ; c) rezultāti

Saskarnes elementi:

a)

1. Teksts” Jautajums/Jautajuma teksts”
2. Atbilžu izvēle
3. Poga “nakamajs jautajums”
4. Poga “Help”

b)

1. Teksts”Treniņa Rezultati”
2. Teksts”Procenti”
3. Teksts”Atzime”
4. Poga “Beigt”
5. Poga “Help”
   1. Skolotaja saskarne “Skolotaja logs”. Saskarnes struktūra: skat. [1.10. att.](#att1_8)

1.10. att. a)Skolotaja logs

a)

Pabeigt 1 Testu Izpildiša

Skolotaja logs

Atļaut 1 testu izpildinašanu

Dot iespēju Trenieties 1.Tests

Apskatīt visu skolēnu atzīmes 1. Tests

Apskatīt visu skolēnu atzīmes 2.Tests

Dot iespēju Trenieties 2.Tests

Atļaut 2 testu izpildinašanu

Pabeigt 2 Testu Izpildiša

dzēst visas skolēnu atbildes

Iziet no sistēmas

Help

Exit

1.Tests

2.Tests

a)

1. Teksts” Skolotaja logs”
2. Teksts” 1.Tests”
3. Teksts” 2.Tests”
4. Poga “Help”
5. Poga “Exit”
6. Poga “Iziet no sistēmas”
7. Poga “Apskatīt visu skolēnu atzīmes 1. Tests”
8. Poga “Apskatīt visu skolēnu atzīmes 2.Tests”
9. Poga “Dot iespēju Trenieties 1.Tests”
10. Poga “Dot iespēju Trenieties 2.Tests”
11. Poga “Atļaut 1 testu izpildinašanu”
12. Poga “Atļaut 2 testu izpildinašanu”
13. Poga “Pabeigt 1 Testu Izpildiša”
14. Poga “Pabeigt 2 Testu Izpildiša”
15. Poga “dzēst visas skolēnu atbildes”
    1. Administratoru logs: skat. [1.11. att.](#att1_8)

a)

Admin Logs

Dzest Kontu

Apskatit

Iziet no sistēmas

Help

Exit

Lietotajvards:

Visi konti

1.11. att. a) Administratoru logs

a)

1. Teksts” Admin logs”
2. Teksts” Lietotajvards:”
3. Teksts” Visi konti”
4. Poga “Help”
5. Poga “Exit”
6. Poga “Iziet no sistēmas”
7. Teksta lauks
8. Teksta panel
9. Poga “Apskatit”
10. Poga “Dzest Kontu”

# Priekšmetiskās jomas klašu sistēmas izstrāde

## Klašu diagramma

2.1. att. Testēšanas sistēmas klašu diagramma UML valodā

Klase CurrentUser ir atbildīga par aktuālā lietotāja datu uzglabāšanu.

username: String – mainīgais, kurā tiek saglabāts lietotājvārds.

role: String – mainīgais, kas norāda lietotāja lomu ("Skolotajs", "Admin", "Student").

Klase MainWindow – nodrošina lietotāja saskarni un rezultātu apstrādi.

atzīme: int – mainīgais, kas uzglabā konkrētu vērtējumu.

filePath: String – mainīgais, kas satur faila ceļu rezultātu saglabāšanai.

atbildes: List<Integer> saraksts ar atbildēm (pareizo atbilžu skaits).

Metodes:

saveResults1(atzīme: int, atbildes: List<Integer>) – metode, kas saglabā rezultātus noteiktā formātā.

saveResults2(atzīme: int, atbildes: List<Integer>) – alternatīva metode rezultātu saglabāšanai cita formata

averageScoreFile(filePath: String): double – metode, kas aprēķina un atgriež vidējo rezultātu no norādītā faila.

## Klašu realizēšana Java valodā

Izstrādātās klašu diagrammas ([2.1. att.](#att2_1)) realizēšanai ***NetBeans*** vidē tika izveidots projekts ar nosaukumu ***TestingSystem***, kurā izstrādātās klases tika aprakstītas ***Java*** valodā. Šeit tiek piedāvāts klašu vispārīgā apraksta (klašu deklarācijas) programmas kods un metožu apraksti.

Pilns avota programmas kods ar klašu metožu definīcijām ir ievietots pielikumā “[Pielikums A.](#_bookmark33) [Klašu sistēma](#_bookmark33)”, bet programmas kods, kas realizē šīs klašu sistēmas testēšanu (klase ar grafisko lietotāja saskarni) ir ievietots pielikumā “[Pielikums B. testa programmas kods](#_bookmark34)”.

public class MainWindow extends javax.swing.JFrame {

public static void saveResults1(int atzime, List<Integer> atbildes) {

StringBuilder result = new StringBuilder();

result.append(CurrentUser.username).append(",")

.append(CurrentUser.role).append(",")

.append(atzime).append(",");

for (int i = 0; i < atbildes.size(); i++) {

result.append(atbildes.get(i));

if (i < atbildes.size() - 1) {

result.append(";");

}

}

try (BufferedWriter writer = new BufferedWriter(

new FileWriter("Results1.txt", true))) {

writer.write(result.toString());

writer.newLine();

} catch (IOException e) {

JOptionPane.showMessageDialog(

null, "Kļūda saglabājot rezultātus: " + e.getMessage());

}

}

public static void saveResults2(int atzime, List<Integer> atbildes) {

StringBuilder result = new StringBuilder();

result.append(CurrentUser.username).append(",")

.append(CurrentUser.role).append(",")

.append(atzime).append(",");

for (int i = 0; i < atbildes.size(); i++) {

result.append(atbildes.get(i));

if (i < atbildes.size() - 1) {

result.append(";");

}

}

try (BufferedWriter writer = new BufferedWriter(

new FileWriter("Results2.txt", true))) {

writer.write(result.toString());

writer.newLine();

} catch (IOException e) {

JOptionPane.showMessageDialog(

null, "Kļūda saglabājot rezultātus: " + e.getMessage());

}

}

public double averageScoreFile(String filePath) {

List<Integer> scores = new ArrayList<>();

try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {

String line;

while ((line = br.readLine()) != null) {

String[] parts = line.split(",");

if (parts.length >= 3) {

try {

int atzime = Integer.parseInt(parts[2]);

scores.add(atzime);

} catch (NumberFormatException e) {

System.out.println("Kļūda: " + parts[2]);

}

}

}

} catch (IOException e) {

e.printStackTrace();

}

if (scores.isEmpty()) {

return 0;

}

int sum = 0;

for (int score : scores) sum += score;

return (double) sum / scores.size();

}

public class CurrentUser {

public static String username = "";

public static String role = "";

}

# Testa programmas izstrāde

## Testēšanas metodikas

Pirms priekšmetiskās jomas klašu sistēmas testēšanas programmas izstrādes, ir jānosaka testēšanas metodika.

Ir jāatšķir koda testēšana un atkļūdošana. Atkļūdošanu veic programmētājs, izmantojot iebūvētos izstrādes vides rīkus un balstoties uz pieredzi programmas koda rakstīšanā. Būtībā runa iet par sintaktisko un semantisko kļūdu identificēšanu programmas tekstā.

Testēšana ir process, kurā nepieciešams plānot un īstenot vairākas sākotnējās procedūras, no kurām galvenā ir testa piemēru kopas izstrāde, kas veido testa plānu. Testpiemēri vairumā gadījumu ir balstīti uz sistēmas funkcionālajām prasībām un var ietekmēt dažādus attīstības līmeņus (vienības testēšana, integrācijas testēšana, sistēmas testēšana).

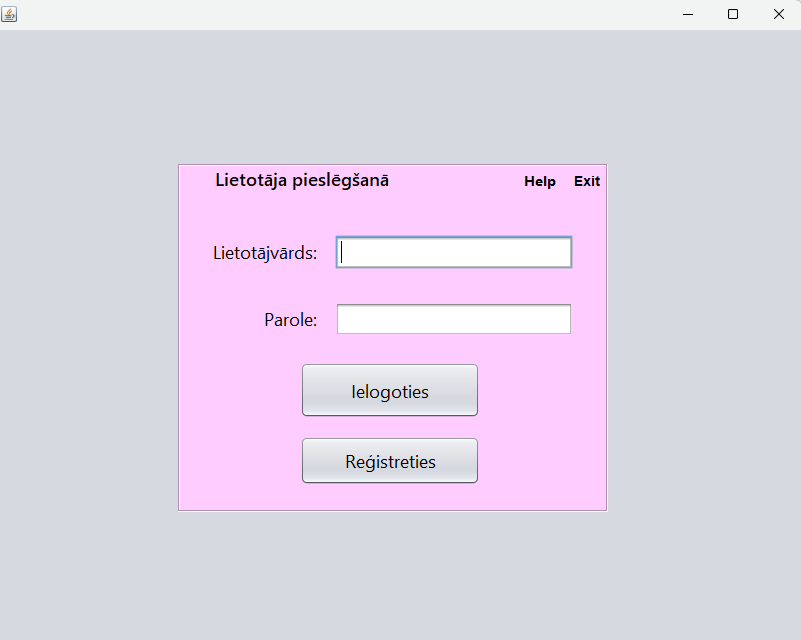
Projekta darba laikā nepieciešams veikt vienkāršotu vienības testēšanas versiju, kas realizē visu izstrādāto klašu metožu testēšanu. Ar testēšanas metodiku projekta darbā pieņem dažādu klases metožu izsaukšanas iespēju sarakstu ar gaidāmajiem rezultātiem un šo testēšanas izsaukumu izpildes kārtību.

## Testēšanas programmas struktūra

* Testēšanas programmas projektēšana un programmatūras izstrāde ir ļoti līdzīga iepriekš aprakstītajam klašu sistēmas izstrādes procesam.
* Jāatzīmē tie momenti un prasības, kas pastāv, izstrādājot testēšanas programmu.
* Testēšanas programmai ir loga interfeiss, kas ietver sevī izvēlnes un interaktīvā interfeisa rīkus, ar kuru palīdzību var pārbaudīt izveidoto klašu sistēmu.
* Interfeisa elementu (pogas, saraksti, ievades lauki, dialoglodziņi, grafiskie elementi utt.) komplekts tiek noteikts, pamatojoties uz iepriekšējā sadaļā aprakstīto testēšanas metodiku.
* Testēšanas programmas interfeisa kvalitātei jābūt vienkāršai lietošanā un interaktīvai. Galvenā prasība ir spēja realizēt testēšanas metodes.

1. Ielogošanas procesa apraksts:

* Ielogošanas process paredzēts reģistrēto lietotāju pievienošanai sistēmā. Lai lietotājs varētu piekļūt testam viņam pēc ir nepieciešams autorizēties. Ielogošanas procesā lietotājs ievada iepriekš reģistrētos datus – lietotājvārdu un paroli (3.1. att. ). Ja lietotāja dati sakrīt ar validācijas datiem, tad lietotājam tiek atvērts programmas galvenais logs (3.3. att.).
* Ielogošanas procesā ir iespējamas vairākas kļūdas – neaizpildīti lauki, neprecīzi ievadīti lietotāja dati.

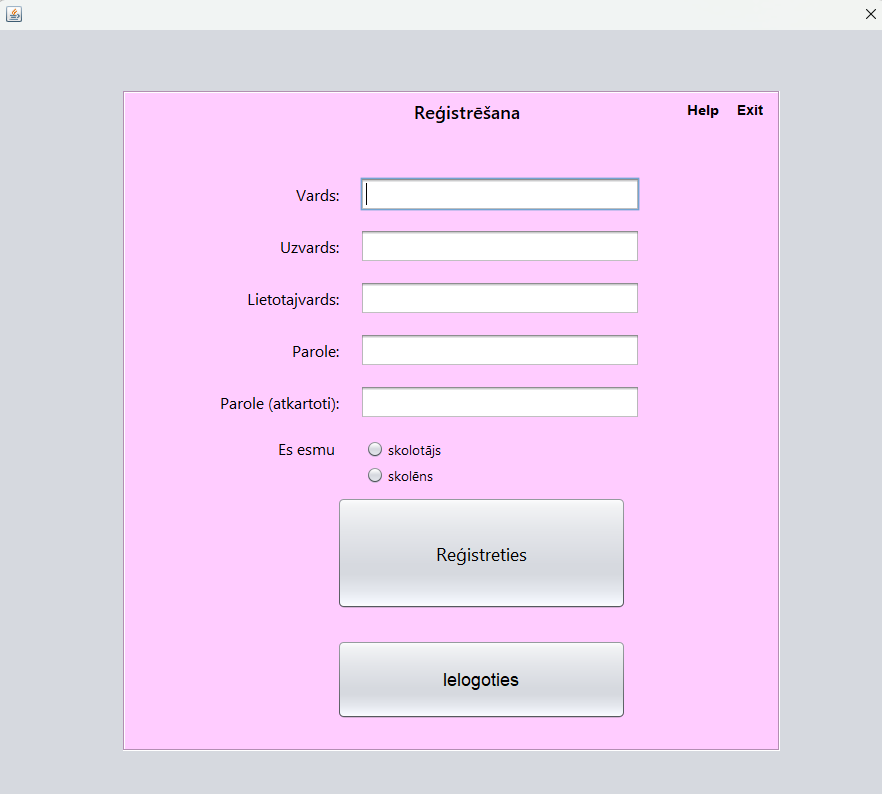


* 1. att. Testēšanas sistēmas ielogošanas forma

1. Reģistrācijas procesa apraksts

• Reģistrācijas process ir paredzēts lietotāju pievienošanai sistēmai. Lai lietotāju reģistrētu sistēmā, lietotājam ir jāsniedz programmā pieprasītie dati. Reģistrācijas procesā lietotājs ievada programmas pieprasītos datus - vārdu, uzvārdu, lietotāja vārdu, paroli , paroles atkārtojumu un nozare(skolotājs vai skolēns) (3.2. attēls). Ja lietotāja dati ir pareizi, viņas datus tiek saglabāti un atveras Testu izveidošanas logs (3.3. attēls).

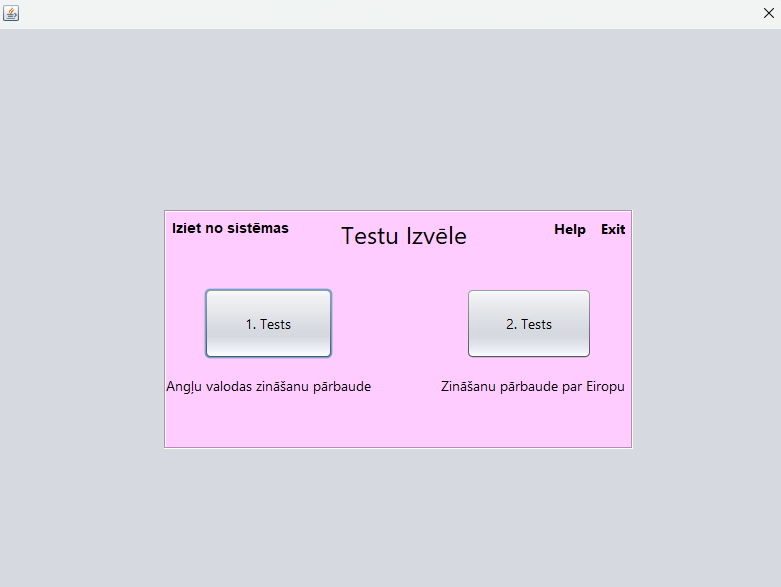
• Reģistrācijas procesā iespējamas vairākas kļūdas - nav aizpildīti lauki, pārsniegts lietotājvārdā un parole garuma ierobežojums ,ja sājš lietotājvārds jau reģistrēts , paroles nesakrīt vai neizvēlas ja lietotājs ir skolotājs vai skolēns.



3.2. att. Testēšanas sistēmas reģistrācijas forma

1. Testa atlases procesa apraksts:

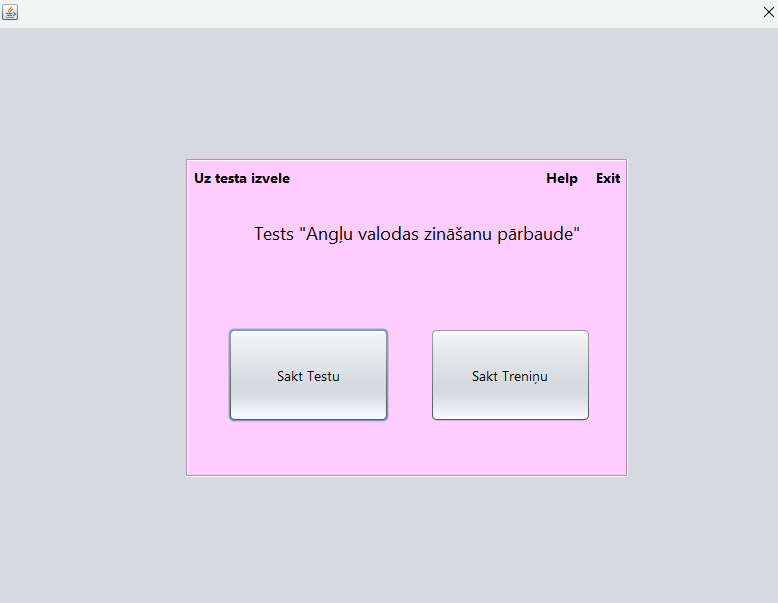
• Testa atlases process ir paredzēts, lai atlasītu un izpildītu testu lietotājiem. Lai sāktu testa izpildi, lietotājam ir jāreģistrējas vai jāpiesakās un jāizvēlas tests (3.3. attēls).



* 1. **att. Testēšanas sistēmas testa formas izvēle.**

1. Testa vai treniņa izvele procesa apraksts:

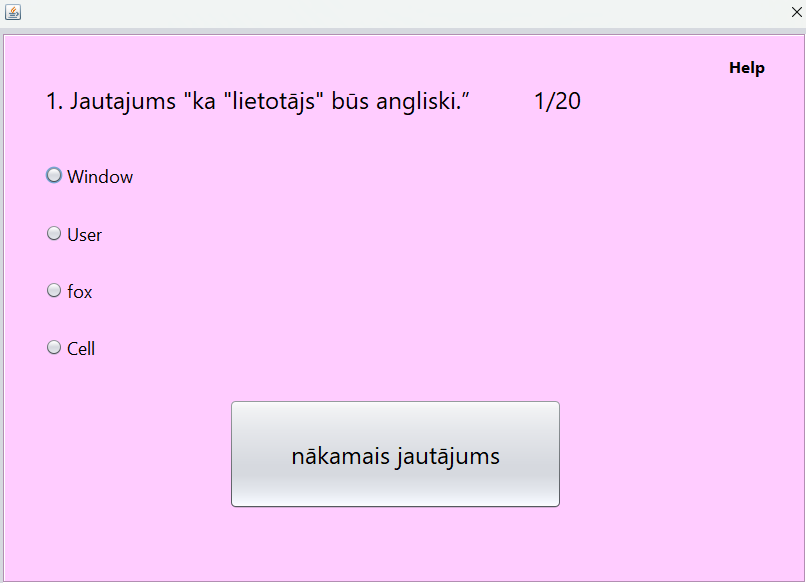
* Testa un Treniņa izvele ir paredzēts, lai atlasītu un izpildītu testu vai treniņu lietotājiem. Lai sāktu testa vai treniņa izpildi, lietotājam ir jāreģistrējas vai jāpiesakās un jāizvēlas tests (3.3. attēls).

****

## att. Testēšanas testa sistēmas formas izvēle.

# Testa izpildes procesa apraksts:

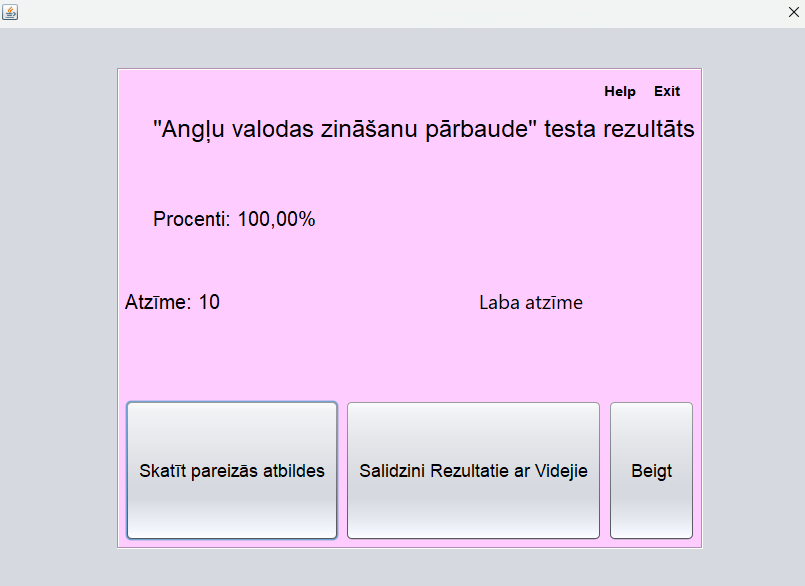
• Testa izpildes process ir izstrādāts tā, lai lietotājs varētu veikt testu un pēc testa veikšanas uzzināt savu rezultātu. Lai nokārtotu vienu no 2 testus, lietotājam jāizpilda tests, kas sastāv no 20 jautājumiem par atsevišķu tēmu. Testa laikā lietotājs izvēlas pareizo atbildi no 4-3 dotajām atbildēm (3.4. attēls). Pēc testas nokārtošanas lietotājam tiek parādīta tabula ar rezultātu un procentiem (3.5. attēls).



* 1. att. Testa formas izpilde testa sistēmai.

# Rezultāta iegūšanas procesa apraksts:

• Rezultātu iegūšanas process ir paredzēts, lai saprastu, cik daudz lietotājs zina par attiecīgo tēmu. Lai iegūtu rezultātu, lietotājam ir jānokārto tests (3.4. attēls), pēc testa nokārtošanas lietotājs var Pabeigt,Apskatīt pareizas atbildes un Paskatit videju atzime klasse nokārtot testu un pabeigt testu (3.5. attēls).



* 1. att. Testa sistēmas testa rezultātu forma.

# Pareizas atbildes skatišana procesa apraksts:

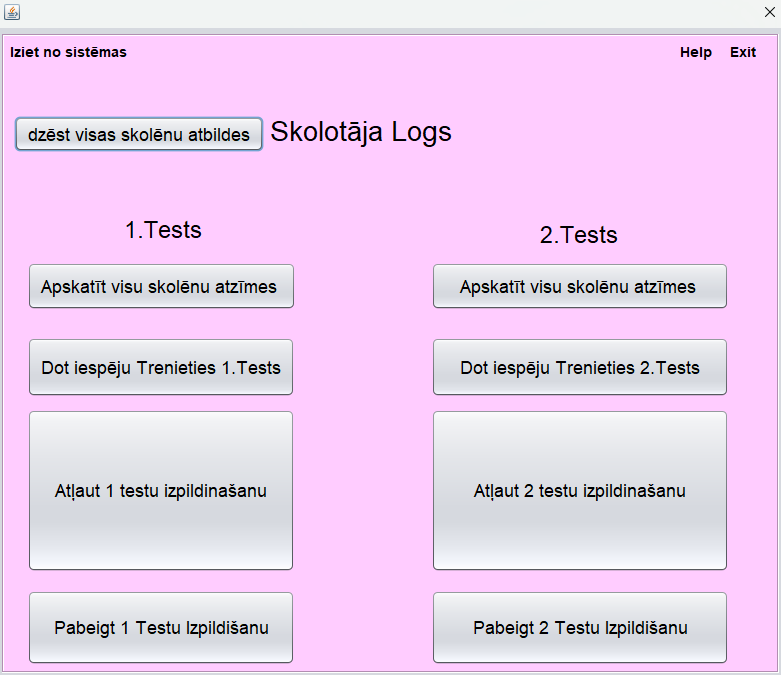
• Pareizas atbildes skatišana procesa ir paredzēts, lai audzeknis, var saprot kadi atbildes biji pareizi par attiecīgo tēmu. Lai varetu paskatit uz pareizas atbildes, lietotājam ir jānokārto tests (3.4. attēls), pēc testa nokārtošanas lietotājs var Apskatīt pareizas atbildes (3.7. attēls).



* 1. att. Pareizu atbildes skatišana sistēmas testa forma.

# Skolotaja logs procesa apraksts:

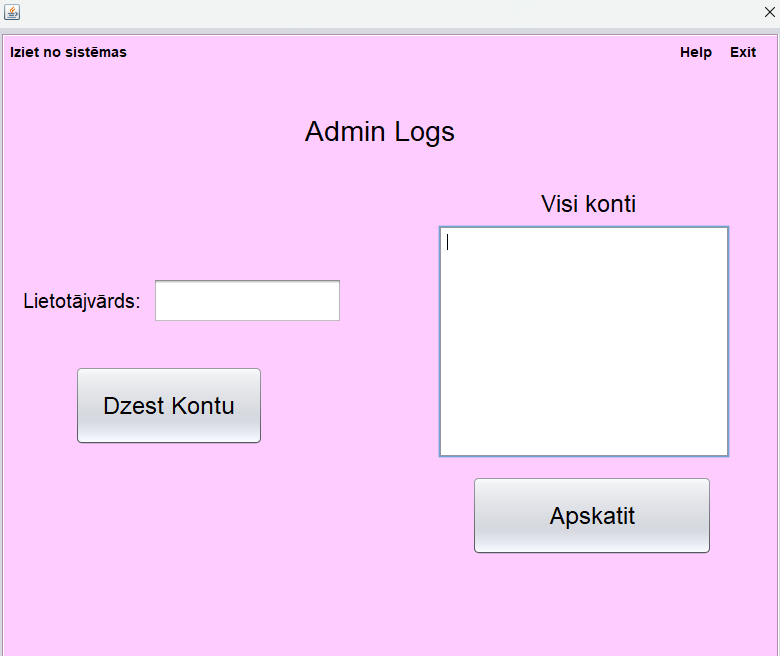
• Skolotaja logs(3.8. att.) ir paredzēts, lai skolotajjs, var atļaut vai pabeigt Testu/Treniņu pildišanu par attiecīgo tēmu. Ari var paskatit visus skolenus atzimes par attiecīgo tēmu, ari var dzest visus skolenus atbildes.



* 1. att. Skolotajā logs forma.

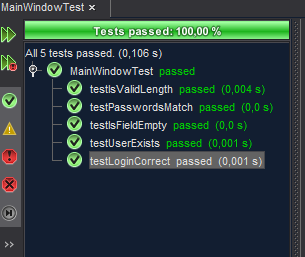
# Administratorā logs procesa apraksts:

• Administratorā logs(3.9. att.) ir paredzēts, lai administators, var paskatit uz visus kontus un izdzest viņem ja administrators grib.



* 1. att. Administratorā logs forma.

## Testēšanas rezultāti



Testēšanas kods testē lietotāju apkalpošanas funkcijas Java valodā. Tiek pārbaudīti:

* vai kāds no ievades laukiem ir tukšs.
* vai lietotājvārds in parole garums ir derigs.
* vai paroles sakrīt.
* vai lietotājs eksistē failā.
* vai lietotājvārds un parole ir pareizi.

visas testēšanas metodes ir izpildītas veiksmīgi — 100% testi ir izturēti (5 no 5).

# Lietotāja rokasgrāmata

## Lietotāja instrukcija

1. Lietotājs palaiž programmu.
2. Piesakās sistēmā, izmantojot savu lietotājvārdu un paroli, ja lietotājam nav sava konta, tad viņš var reģistrēt to.
   1. Reģistrācijas logā lietotājs aizpilda visus nepieciešamus datus un noklikšķina uz pogas Reģistrēties.
3. Lietotāja izvēlnē ir 5 pogas.
   1. Poga «1. Tests» lai sākt pildīt 1 testu.
   2. Poga «2. Tests» lai sākt pildīt 2 testu.
   3. Poga «Help» lai iegūtu palīdzību.
   4. Poga «Exit» lai iziet no programmas.

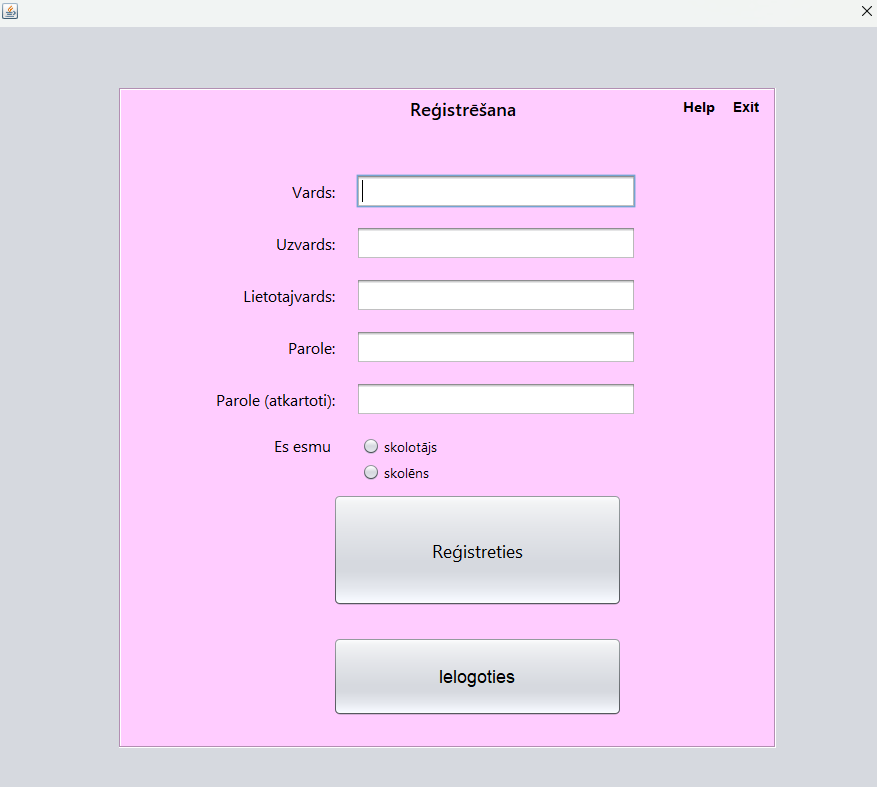
* Pēc testa izpildīšanas tiek paradīts testa izpildes rezultāti.
  1. Poga « Iziet no sistēmas » ļauj iziet no sava konta.

1. Lietotājs testa/treniņa izvele ir 5 pogas .
   1. Poga « Sakt Testu» lai sākt Testu.
   2. Poga « Sakt Treniņu» lai sakt Treniņu.
   3. Poga «Help» lai iegūtu palīdzību.
   4. Poga «Exit» lai iziet no programmas.
   5. Poga « Iziet no sistēmas » ļauj iziet no sava konta.
2. Lietotājs testa ir 3 pogas un 1 izvele .
   1. Poga « nākamais jautājums» lai liet uz nakamu jautajumu vai «pabeidziet testu» .
   2. Poga «Help» lai iegūtu palīdzību.
   3. Poga «Exit» lai iziet no programmas.
   4. Izvele «Jautajumi» lai varētu izvelēties 1 no piedāvātam jautājumam.
3. Lietotājs Rezultai ir 4 pogas .
   1. Poga « Skatīt pareizās atbildes» lai paskatīt pareizās atbilde.
   2. Poga « Salidzini Rezultatie ar Videjie » lai salidzini savu rezultatie ar videjie.
   3. Poga « Beigt » lai pabeigt testu.
   4. Poga « Help » lai iziet no programmas.
   5. Poga «Exit» lai iziet no programmas.
4. Skolotaja izvēlnē ir 12 pogas.
   1. Poga « dzēst visas skolēnu atbildes» lai izdzest visas skolēnu atbildes.
   2. Poga « Apskatīt visu skolēnu atzīmes 1.Tests» lai apskatīt visu skolēnu atzīmes par 1.Testu.
   3. Poga « Apskatīt visu skolēnu atzīmes 2.Tests » lai apskatīt visu skolēnu atzīmes par 2.Testu.
   4. Poga « Dot iespēju Trenieties 1.Tests» lai dot iespēju trenieties 1.Testa.
   5. Poga « Dot iespēju Trenieties 2.Tests » lai dot iespēju trenieties 2.Testa.
   6. Poga « Atļaut 1 testu izpildinašanu» lai atļaut 1 testu izpildinašanu.
   7. Poga « Atļaut 2 testu izpildinašanu» lai atļaut 2 testu izpildinašanu.
   8. Poga « Pabeigt 1 Testu Izpildišanu» lai pabeigt 1 Testu/Treniņu izpildišanu.
   9. Poga « Pabeigt 2 Testu Izpildišanu» lai pabeigt 2 Testu/ Treniņu izpildišanu.
   10. Poga « Iziet no sistēmas » ļauj iziet no skolotaja konta.
   11. Poga «Help» lai iegūtu palīdzību.
   12. Poga «Exit» lai iziet no programmas.
5. Admina izvēlnē ir 5 pogas.
   1. Poga «Exit» lai iziet no programmas.
   2. Poga « Iziet no sistēmas » ļauj iziet no skolotaja konta.
   3. Poga «Help» lai iegūtu palīdzību.
   4. Poga « Apskatit» lai apskatit vius kontus.
   5. Poga « Dzest Kontu» lai dzest kontu.

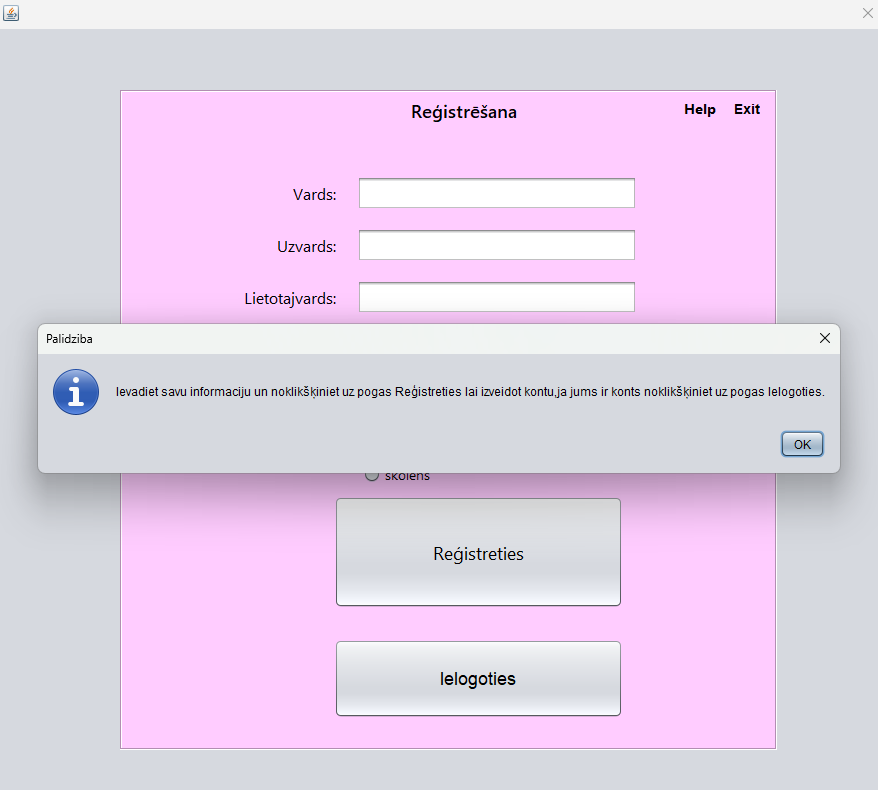
## Palīdzības sistēma (Help)

Lai sāktu pildīt testu, ir jāpiesakās savā kontā, izmantojot lietotājvārdu un paroli. Ja jums nav konta programmā, jums jāreģistrējas, norādot vārdu, uzvārdu, lietotājvārdu un paroli. Pēc tam jums tiks dota iespēja izvēlēties testu vai treniņu un to izpildīt. Varat izvēlēties no 2 testiem: 1. ‘’Angļu valodas zināšanu pārbaude’’ un 2. ‘’Zināšanu pārbaude par Eiropu’’. Jūs izvēlaties vienu no testiem, un jūs varat izpildīt treniņu ja skolotājs atļaut vai sākt pildīt testu, kura augšpusē tiek parādīts jautājums, bet apakšā – 4-3 atbilžu varianti. Jūsu uzdevums ir izvēlēties vienu no atbilžu variantiem un nospiest pogu “nākamais jautājums”. Pabeidzot pildīt testu/treniņu, jūs saņemsiet savu rezultātus (atzīmē/procentus), kas tiks saglabāts datu bāzē.

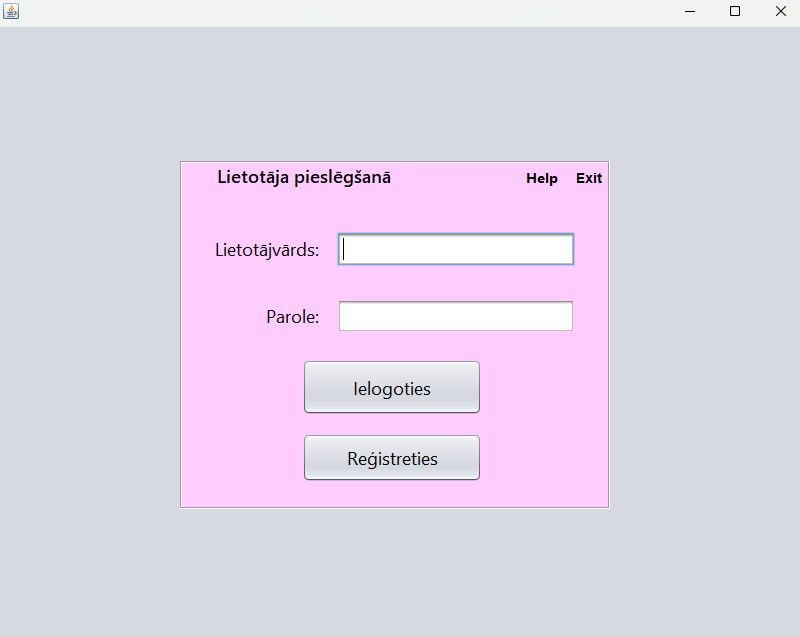
Ja rodas problēmas ar reģistrāciju, lietotājs var noklikšķināt uz pogas “Help” (att. 4.1). Atvērsies dialoglogs, kurā būs aprakstīti reģistrācijas noteikumi (4.2. att.).



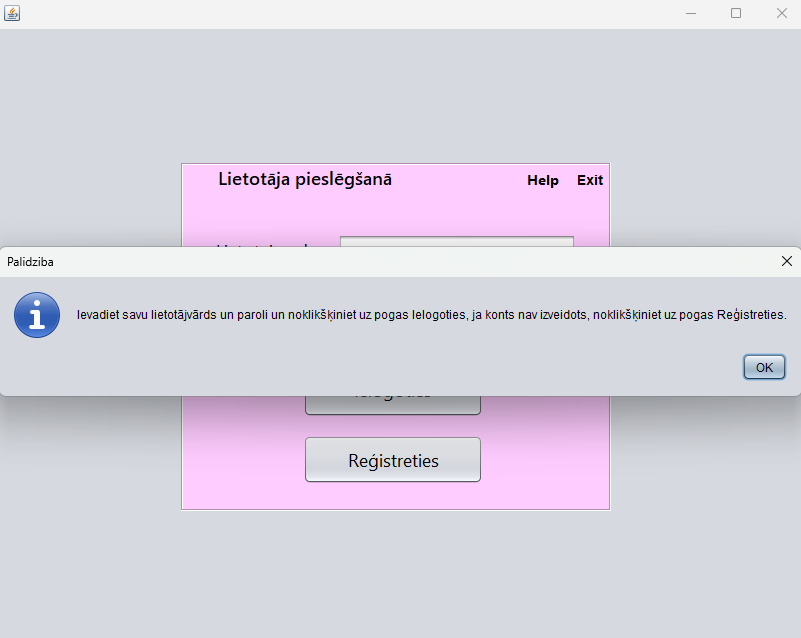
4.1. att. Pogas “Help” piemērs reģistrācijas



4.2. att. Reģistrācijas Help



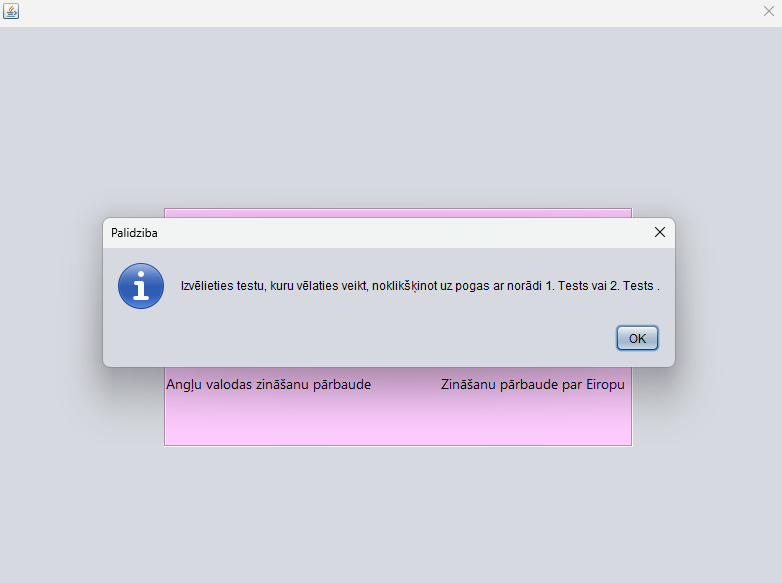
**4.3. att. Pogas “Help” piemērs** **ielogošana**



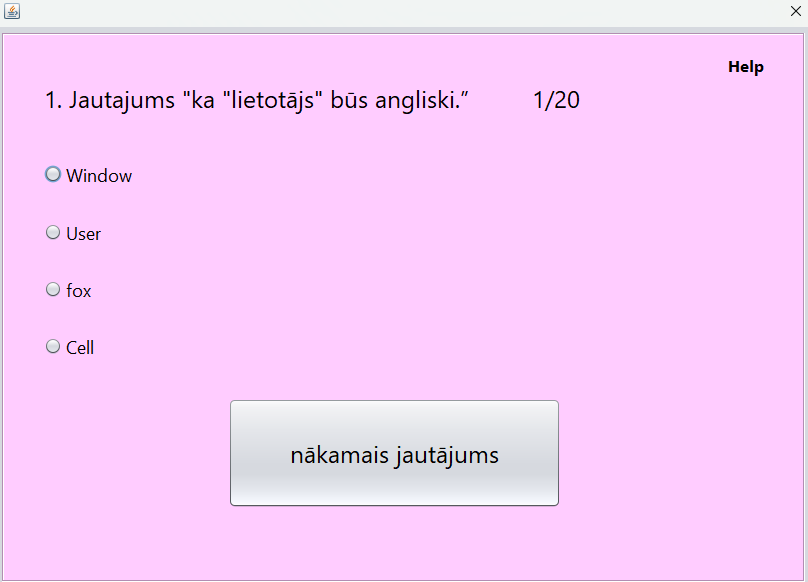
4.4. att. Ielogošana Help



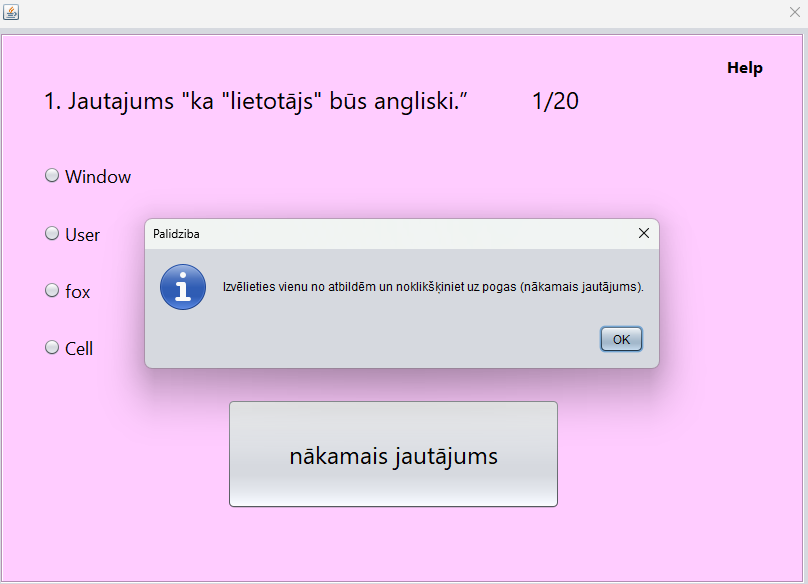
* 1. **att. Pogas “Help” piemērs testu izvelēšanā**

****

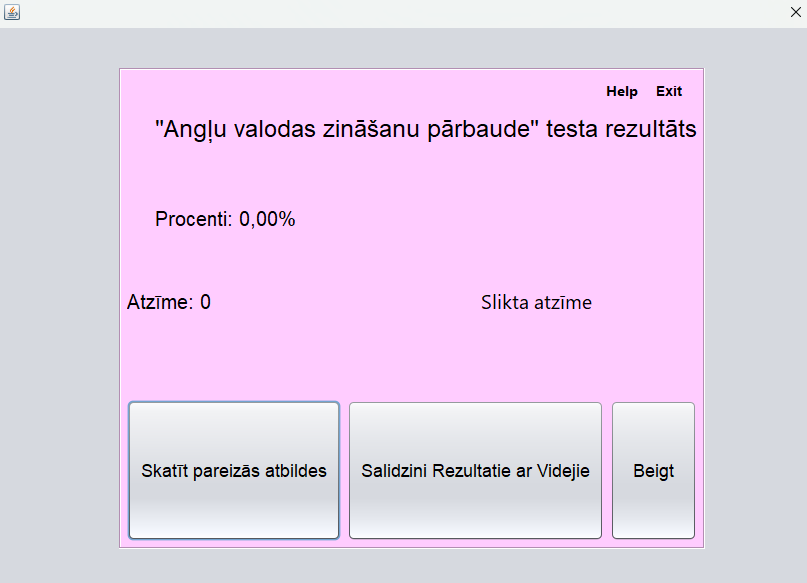
4.6. att. Testu izvelēšanā Help



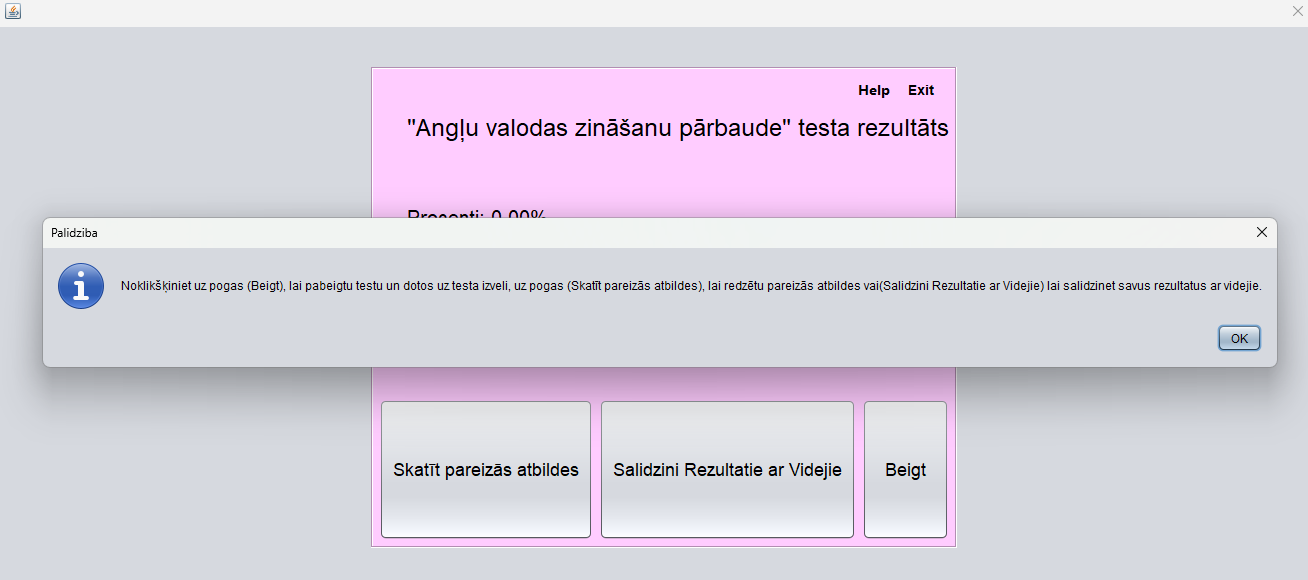
**4.7.att. Pogas “Help” piemērs testa**



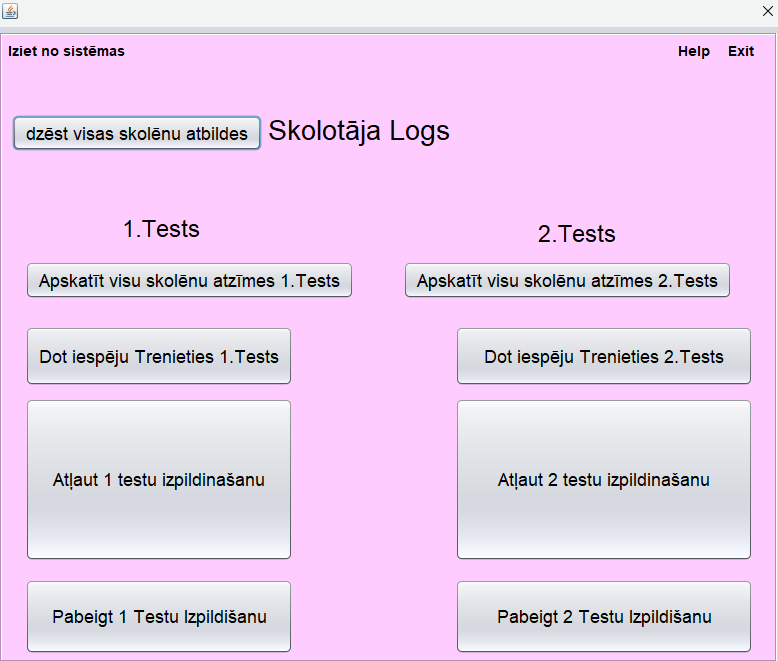
4.8. att. Testu Help



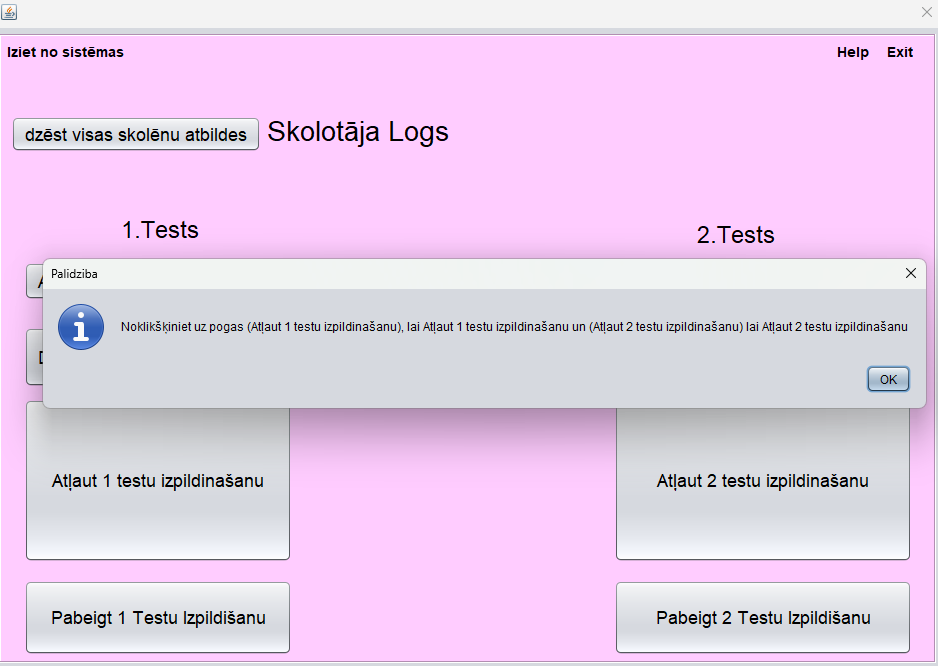
4.9. att. Pogas “Help” piemērs rezultatiem



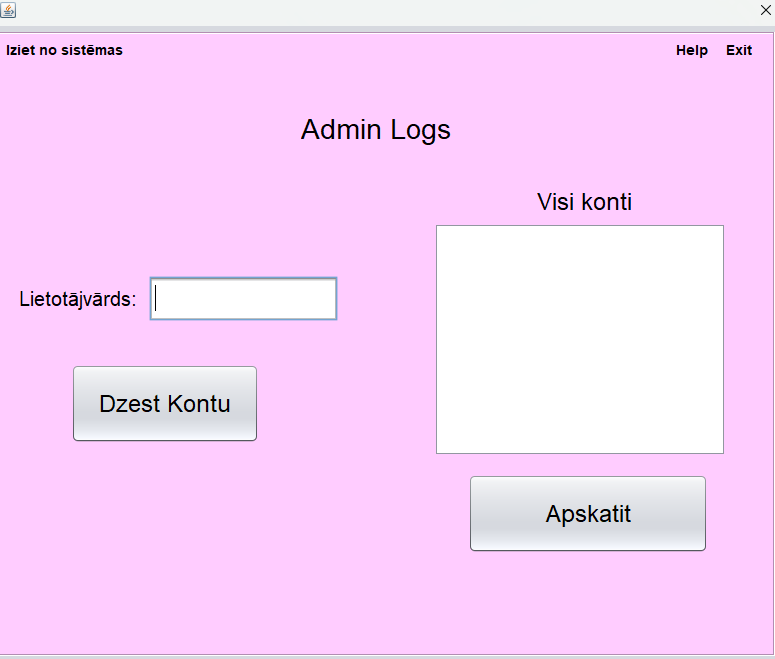
**4.10.att. Rezultātiem** **Help**



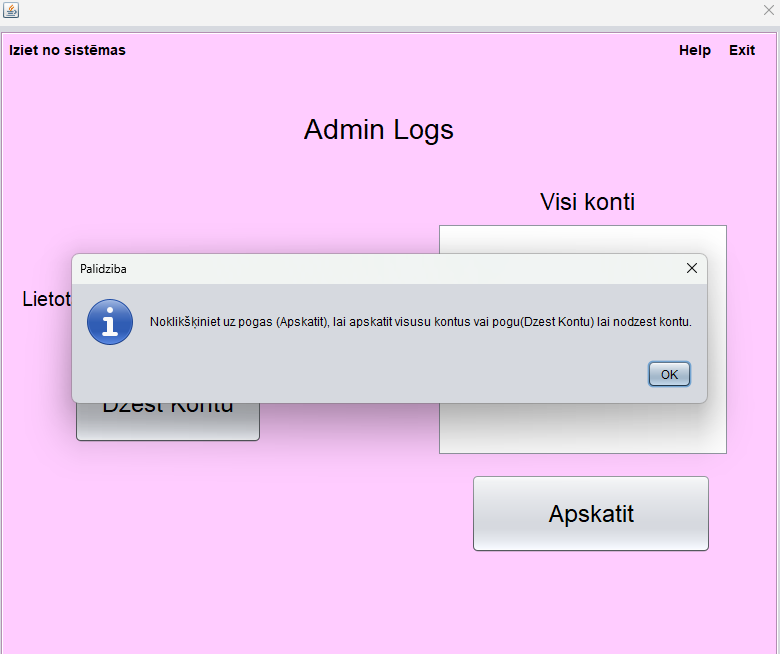
4.11. att. Pogas “Help” piemērs skolotajā logs



**4.12.att. Skolotaja logs Help**



4.13. att. Pogas “Help” piemērs administratora logs



**4.14.att. Administratora logs Help**

# Secinājumi

Tika izstrādāts projekts «Testēšanas sistēma», izmantojot Java valodu un Swing grafisko vidi. Projekts ietver reģistrācijas, autorizācijas, testu veikšanas un rezultātu saglabāšanas funkcionalitāti. Izstrādes gaitā nācās risināt dažādas problēmas, īpaši saistītas ar failu apstrādi un lietotāja datu validāciju. Darbs bija interesants un deva iespēju apgūt jaunas zināšanas par GUI izveidi, darbībām ar failiem un lietotāju datu pārvaldību. Lai gan dažas idejas netika realizētas, svarīgākās funkcijas ir pabeigtas un projekts veiksmīgi darbojas. Šī bija vērtīga pieredze, kas noderēs turpmākajiem darbiem. Ceru, jo tālāk, jo labāk!

# Izmantotās literatūras (informācijas avotu) saraksts

1. **Zaiceva, L.** *Programmatūras izstrādes tehnoloģija.* Rīga : RTU, 2002.

2. **Kodors, Sergejs.** *IEVADS PRASĪBU INŽENIERIJĀ. Mācību līdzeklis.* Rēzekne : Rēzeknes Tehnoloģiju akadēmija, 2019. gada. 978-9984-44-226-6.

3. **Dortiņa, I. un Dortiņš, A.** 1.1.1. Ievads Java. Java pamati. [Tiešsaiste] 2024. gada 13. 2. [Citēts: 2025. gada 20. 3.] https://skolo.lv/mod/resource/view.php?id=82126301.

4. **—.** 1.1.2. Java sazarotas struktūras operātori. [Tiešsaiste] 2024. gada 14. 2. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82257905.

5. —. 1.1.3. Java cikliskās struktūras operatori. [Tiešsaiste] 2022. gada 22. 2. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82383944.

6. —. 1.2.1.1. Java masīvi. [Tiešsaiste] 2024. gada 29. 2. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82503734.

7. —. 1.2.1.2. Java rakstzīmju masīvi un rakstzīmju virknes. [Tiešsaiste] 2024. gada 7. 3. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82608403.

8. —. 1.2.1.3. Java dinamiskie masīvi. [Tiešsaiste] 2024. gada 13. 3. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82703677.

9. —. 1.2.2. Java ieraksti. [Tiešsaiste] 2024. gada 19. 3. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82785148.

10. —. 1.2.3. Java Ievades/izvades sistēma. [Tiešsaiste] 2024. gada 3. 4. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=82944472&redirect=1.

11. —. 2.1.1.1. Java. OOP koncepcija. [Tiešsaiste] 2024. gada 16. 4. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=83119822.

12. **Dortiņa, I. un Stašanova, L.** 2.1.1.2. Java. Metodes un konstruktori. [Tiešsaiste] 2024. gada 2. 9. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=84426472.

13. **—.** 2.1.2. Java. Mantošana un metožu pārdefinēšana. [Tiešsaiste] 2024. gada 30. 9. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=85417326.

14. —. 2.1.3. Java pakotnes un interfeisi. [Tiešsaiste] 2024. gada 14. 10. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=86185154.

15. —. 2.2. Java izņēmumi un to apstrāde. [Tiešsaiste] 2022. gada 23. 10. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=86272136&redirect=1.

16. —. 4.1.1. Java. Grafiskais lietotāja interfeiss. [Tiešsaiste] 2025. gada 23. 1. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=87282004.

17. —. 4.1.2. Java GUI izveidošana NetBeans vidē. [Tiešsaiste] 2025. gada 5. 2. [Citēts: 2025. gada 20. 2.] https://skolo.lv/mod/resource/view.php?id=87334441.

18. Latvijas Nacionālais terminoloģijas portāls. *termini.gov.lv.* [Tiešsaiste] Izstrādātājs: Tilde., 2005-2023. gada. [Citēts: 2025. gada 20. 2.] https://termini.gov.lv/.

19. **Taylor Johnson, Dung X. Nguyen.** UML and more JAVA Syntax. [Tiešsaiste] COMP 212 LAB 1.5, 2007. gada 16. 1. [Citēts: 2025. gada 20. 2.] https://www.clear.rice.edu/comp212/07-spring/labs/01.5/.

# Pielikums A. Klašu sistēma

public class Teacher extends User {

public Teacher(String fullName, String login, String password) {

super(fullName, login, password);

}

public void StartTest() {}

public void reviewAnswers() {}

public void StartTraining() {}

}

public class CurrentUser {

public static String username = "";

public static String role = "";

}

public class DistanceExaminer {

private List<User> users = new ArrayList<>();

private List<Question> questionBank = new ArrayList<>();

public User currentUser;

public void registerUser(String name, String login, String password, String role) {

if (role.equalsIgnoreCase("teacher")) {

users.add(new Teacher(name, login, password));

} else {

users.add(new Student(name, login, password));

}

}

public User loginUser(String login, String password) {

for (User user : users) {

if (user.authenticate(login, password)) {

currentUser = user;

return user;

}

}

return null;

}

public List<Student> getAllStudentResults() {

List<Student> results = new ArrayList<>();

for (User user : users) {

if (user instanceof Student) {

results.add((Student) user);

}

}

return results;

}

public List<Double> getAllScores() {

List<Double> scores = new ArrayList<>();

for (Student s : getAllStudentResults()) {

scores.add(s.User());

}

return scores;

}

public void saveToFile() {

try (BufferedWriter writer = new BufferedWriter(new FileWriter("users.txt"))) {

for (User user : users) {

String role = (user instanceof Teacher) ? "teacher" : "student";

writer.write(user.getFullName() + ";" + user.getLogin() + ";" + user.getPassword() + ";" + role);

writer.newLine();

}

} catch (IOException e) {

e.printStackTrace();

}

}

public void loadFromFile() {

users.clear();

try (BufferedReader reader = new BufferedReader(new FileReader("users.txt"))) {

String line;

while ((line = reader.readLine()) != null) {

String[] parts = line.split(";");

if (parts.length == 4) {

registerUser(parts[0], parts[1], parts[2], parts[3]);

}

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

public class Question {

private String prompt;

private String correctAnswer;

public Question(String prompt, String correctAnswer) {

this.prompt = prompt;

this.correctAnswer = correctAnswer;

}

public String getPrompt() {

return prompt;

}

public boolean isCorrect(String answer) {

return correctAnswer.equalsIgnoreCase(answer.trim());

}

public String getCorrectAnswer() {

return correctAnswer;

}

}

public class Student extends User {

private int totalQuestions = 0;

private int correctAnswers = 0;

public Student(String fullName, String login, String password) {

super(fullName, login, password);

}

public void submitAnswer(Question question, String givenAnswer) {

totalQuestions++;

if (question.isCorrect(givenAnswer)) {

correctAnswers++;

}

}

public void resetScore() {

totalQuestions = 0;

correctAnswers = 0;

}

public double calculateScore() {

return totalQuestions == 0 ? 0.0 : (double) correctAnswers / totalQuestions;

}

Double User() {

throw new UnsupportedOperationException("Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody

}

}

public class User {

protected String fullName;

protected String login;

protected String password;

public User(String fullName, String login, String password) {

this.fullName = fullName;

this.login = login;

this.password = password;

}

public boolean authenticate(String login, String password) {

return this.login.equals(login) && this.password.equals(password);

}

public String getFullName() {

return fullName;

}

public String getLogin() {

return login;

}

public String getPassword() {

return password;

}

}

# Pielikums B. testa programmas kods

.

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import org.junit.AfterClass;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class MainWindowTest {

private static File tempLoginFile;

@BeforeClass

public static void setup() throws IOException {

tempLoginFile = File.createTempFile("LoginPassword", ".txt");

try (BufferedWriter writer = new BufferedWriter(new FileWriter(tempLoginFile))) {

writer.write("Gunars,Gravitis,Alien,123,student\n");

}

}

@AfterClass

public static void teardown() {

if (tempLoginFile.exists()) {

tempLoginFile.delete();

}

}

@Test

public void testIsFieldEmpty() {

assertTrue(UserService.isFieldEmpty("qwe", "", "321"));

assertFalse(UserService.isFieldEmpty("qwe", "321", "zxc"));

}

@Test

public void testIsValidLength() {

assertTrue(UserService.isValidLength("Gunars", 3, 15));

assertFalse(UserService.isValidLength("zx", 3, 15));

assertFalse(UserService.isValidLength("SUPER\_SUPER\_PUPER\_LOOONG\_NAME", 3, 15));

}

@Test

public void testPasswordsMatch() {

assertTrue(UserService.passwordsMatch("123", "123"));

assertFalse(UserService.passwordsMatch("123", "Alien123"));

}

@Test

public void testUserExists() throws IOException {

assertTrue(UserService.userExists("Alien", tempLoginFile));

assertFalse(UserService.userExists("Megaladon", tempLoginFile));

}

@Test

public void testLoginCorrect() throws IOException {

assertTrue(UserService.loginCorrect("Alien", "123", tempLoginFile));

assertFalse(UserService.loginCorrect("Megaladon", "Alien123", tempLoginFile));

}

public class UserService {

public static boolean isFieldEmpty(String... fields) {

for (String field : fields) {

if (field == null || field.trim().isEmpty()) return true;

}

return false;

}

public static boolean isValidLength(String input, int min, int max) {

return input != null && input.length() >= min && input.length() <= max;

}

public static boolean passwordsMatch(String pass1, String pass2) {

return pass1 != null && pass1.equals(pass2);

}

public static boolean userExists(String username, File file) throws IOException {

try (BufferedReader reader = new BufferedReader(new FileReader(file))) {

String line;

while ((line = reader.readLine()) != null) {

String[] parts = line.split(",");

if (parts.length > 2 && parts[2].trim().equals(username.trim())) {

return true;

}

}

}

return false;

}

public static boolean loginCorrect(String username, String password, File file) throws IOException {

try (BufferedReader reader = new BufferedReader(new FileReader(file))) {

String line;

while ((line = reader.readLine()) != null) {

String[] parts = line.split(",");

if (parts.length >= 4 &&

parts[2].trim().equals(username) &&

parts[3].trim().equals(password)) {

return true;

}

}

}

return false;

}

}

}