Latvijas Republikas izglītības un zinātnes ministrija

Profesionālās izglītības kompetences centrs

Daugavpils tehnikums

**PRAKTISKAIS DARBS**

Programmēšanas tehniķis

specialitāte

Praktiskais darbs Nr.1

temats

Lineāras programmas

PASKAIDROJOŠAIS RAKSTS

**DT.P.D.0018.018PA**

Profesijas kods 33 481 031 Programmēšanas nodaļa

PR-21. grupa

Izveidoja E. Sinkevičs

paraksts V.,Uzvārds

Pārbaudīja I. Dortiņa

paraksts V.,Uzvārds

2020./2021. m.g.

**Saturs**

[Ievads 4](#_Toc102472961)

[1. Uzdevuma nostādne 5](#_Toc102472962)

[1.1. Sistēmas modelis 5](#_Toc102472963)

[1.1.1. Priekšmetiskās jomas informācijas modelis 5](#_Toc102472964)

[1.1.2. Sistēmas apkārtnes shēma 5](#_Toc102472965)

[1.1.3. Viedokļu analīze 5](#_Toc102472966)

[1.1.4. Datu modelēšana 5](#_Toc102472967)

[1.1.5. Datu vārdnīca 5](#_Toc102472968)

[1.2. Sistēmas evolūcija 5](#_Toc102472969)

[1.3. Sistēmas funkcionālās prasības 5](#_Toc102472970)

[1.4. Sistēmas nefunkcionālās prasības 5](#_Toc102472971)

[2. Priekšmetiskās jomas klašu sistēmas izstrāde 6](#_Toc102472972)

[2.1. Klašu diagramma 6](#_Toc102472973)

[2.2. Klašu realizēšana java valodā 6](#_Toc102472974)

[3. Testa programmas izstrāde 7](#_Toc102472975)

[3.1. Testēšanas metodikas 7](#_Toc102472976)

[3.2. Testēšanas programmas struktūra 7](#_Toc102472977)

[3.3. Testēšanas rezultāti 7](#_Toc102472978)

[4. Lietotāja rokasgrāmata 7](#_Toc102472979)

[4.1. Lietotāja instrukcija 7](#_Toc102472980)

[4.2. Palīdzības sistēma (help) 7](#_Toc102472981)

[4.3. Sistēmas ziņojumi 7](#_Toc102472982)

[5. Secinājumi 8](#_Toc102472983)

[6. Izmantotās literatūras (informācijas avotu) sarakst 9](#_Toc102472984)

[7. Pielikums a. klašu sistēma 10](#_Toc102472985)

[8. Pielikums b. testa programmas kods 11](#_Toc102472986)

# Ievads

# 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.**).

Datu bāze

Tests

Rezultāti

Administrators/Skolotājs

Lietotājs

1.1.att. Vienkāršots sistēmas modelis

Saišu nozīme vienkāršotā nodaļas modelī:

• Skolotājs ir testa veidotājs un pārbaudītājs, kuram ir pieeja pie atbildēm, testam un lietotāja rezultātiem;

• Datu bāze satur failus ar jautājumiem un ar pariezām atbildēm;

• Lietotājs ir 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, pareizās un nepareizās atbildes skaitliski un procentuāli.

### Sistēmas apkārtnes shēma

Dati, kā atbildes, ko ievada audzēknis tiek saglabāti failā. Tomēr dažreiz atbildes netiek automātiski novērtētas un skolotājam ir iespēja manuāli novērtēt iesniegto atbildi. Saskarni ar lietotājiem nodrošinās interfeisa modulis (**att. 1.2**).

Lietotājs

Lietotāja saskarne

Sistēma

Datu bāze

Lietotāja dati un rezultāti

att. 1.2. Sistēmas apkārtnes modeļa piemērs

### Viedokļu analīze

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

Funkcionālie viedokļi

Datu viedokļi

Viedokļi par lietotāju

Nefunkcionālie viedokļi

att. 1.3. Viedokļu burbuļu diagramma

Otrais etaps viedokļu analīzē ir viedokļu klasifikācija, kad viedokļi ir sagrupēti grupās (**att. 1.4**).

Testēšanas

sistēma (V0)

Lietotājs (V11)

Datu bāze (V12)

Tests (V13)

Skolotājs (V111)

Lietotāju registrs (V121)

Audzekņa funkcijas (V132)

Skolotāja funkcijas (V131)

Audzēknis (V112)

Testa jautajumu registrs (V122)

Testa izveide (V1311)

Testa pildīšana (V1321)

Testa rezultātu saņemšana (V1322)

Testa manuāla novērtēšana (V1312)

att. 1.4. Viedokļu hierarhiskā struktūra

### Datu modelēšana

Datu modelēšanai ir tiek izmantots ER modelis. Testēšanas sistēmas ER modelis (**att. 1.5**) 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

Satur

Skolotājs

Izveido

Jautājums

att. 1.5. Vienkārsots testēšanas sistēmas ER modelis

### Datu vārdnīca

|  |  |  |
| --- | --- | --- |
| Datu tips | Nosaukums | Apraksts |
| String | **name** | Audzēknis ievada savu vārdu un uzvārdu |
| String | **questionText** | Neliels teksta apgabals, kas apraksta jautājumu |
| String[] | **answersText** | Atbilžu variantu masīvs. Tiek piedāvāti vairāki atbilžu varianti, kur ir jāizvēlas viens pareizs |
| int | **correctAnswer** | Aile, kas nosaka, vai tika izvēlēts pareizs atbildes variants |
| float | **result** | Audzēkņa saņemto punktu skaits (0-100) |
| int | **mark** | Audzēkņa atzīme (0-10) |

## 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 – ielogosanai sistēmā un trešā – testa pildīšanai. Grafisko lietotaja saskarni var pilnveidot, papildinot to ar dažātām krasu un noformejuma shemām. Testēšanas sistemā paredzēti divu lietotaju veidi: administrators un lietotajs, kurš pilda testu. Nākotnē plānots pievienot vēl vienu lietotāju grupu ar funkciju veidot jaunus testus un pārbaudīt izpilītos testus, līdz ar to lietotājiem būs bivas lomas: Audzeknis, kurš pilda testu un skolotajs, kurš veido un pārbauda testus. Administratoram plānots realizēt funkciju mainīt lietotāja vārdu, uzvārdu, lietotājvārdu vai paroli, ka arī pievienot un dzēst lietotājus un mainīt vinu lomas. Programma satur vienu testu, kurā ir 10 jautajumi, tomēr nākotnē sistēmu var pilnveidot, pievienojot jaunas testa tēmas, kļūdu uzrādīšanu un kļūdu skaidrojumu pēc testa izpildes, lietotāju reitingu, lietotāju datu izvadi lietotāja izvelnē vai administratora izvelnē. Sistēmas dažiem datu tipiem tiks izmantoti dati, kuri tiek saglabati teksta failā, nākotnē programmu var pilnveidot darbam ar dau bāzi (datu bāzes savienojums, testa jautājumi un atbildes uz tiem u.c.)

## Sistēmas funkcionālās prasības

**1. Funkcija “Reģistrēties sistemā”**

***ID: T1***

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

**Ievade:**

1) Lietotāja vārds

2) Lietotāja uzvārds

3) Lietotāja lietotājvārds

4) Lietotāja parole

5) Ievadītās paroles apstiprināšana.

**Apstrāde:**

1) Lietotāja vārda saglābašana failā.

2) Lietotāja uzvārda saglābašana failā.

3) Lietotāja lietotājvārda saglabāšana failā.

4) Lietotāja paroles saglabāšana failā.

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

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

**2. Funkcija “Ielogoties sistemā”**

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

**3. 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!"

**4. Funkcija “Pildīt testu”**

***ID: 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 parādīts dialoglodziņš ar uzrakstu "Veiksmi testā!"

**5. Funkcija “Atbildēt uz testa jautajumu”**

***ID: T5***

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

**Ievade:** Lietotājs izvēlas 1 no 4 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", pēc kura tiek parādīts nākamā izvēlne ar testa jautājumiem.

**6. Fukcija “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.

## Sistēmas nefunkcionālās prasības

1. Prasības produktam:

1.1. Lietotāju saskarne ar sistēmu notiek latviešu valodā.

1.2. Produktam nav paredzētas vairākas sastāvdaļas. Produktam ir trīs ārējās saskarnes: reģistrēšanai sistēmā, ielogosanai sistēmā, testa pildīšanai.

1.3. Programma ir paredzēta vienam lietotājam, kurš saprot latviešu valodu. Lietotāja vecuma vai izglītības ierobežojumi nav paredzēti.

2. Ārējās saskarnes prasības

2.1. Lietotāja saskarne: produkts paredz vienotu lietotāja saskarni.

2.2. Ekrāna formāti: minimālais ekrāna izmērsir 10” ar minimālo izšķirtspēju –800x600 punkti

2.3. Lietotāja saskarne “Lietotāja reģistrāšanas saskarne”. Saskarnes struktūra: skat. (**1.6. att).**

Vārds:

Uzvārds:

Lietotājvārds:

Parole:

Parole (atkārtoti):

Reģistrēties

1

2

3

4

5

6

**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. poga “Reģistrēties”

2.4. Lietotāja saskarne “Lietotāja ielogošanas saskarne”. Saskarnes struktūra: skat. (**1.7. att).**

Lietotājvārds:

Parole:

Reģistrēties

1

2

3

4

Ielogoties

**1.7. att. Lietotāja ielogošanas saskarne**

Saskarnes elementi:

1. teksta lauks “Lietotājvārds”

2. teksta lauks ‘Parole”

3. poga “Ielogoties”

4. poga “Reģistrēties”

2.5. Lietotāja saskarne “Testa izpilde”. Saskarnes struktūra: skat. (**1.8. att**).

c)

b)

a)

“Tests”

Pildīt testu

1.uzd “Jautājums”

Atbilde 1

Atbilde 2

2

Atbilde 3

Tālak

Rezultāti, pareizo atbilžu skaits, procenti,

Atzīmes izvade…

Atzīme: X

Procenti: Y%

Beigt

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

Saskarnes elementi:

**a)**

1) Uzraksts “*Tests*”

2) poga “*Pildīt testu*”

**b)**

1)Uzraksts *1.uzd “Jautājums”*

2) 1 testa atbilde

3) 2 testa atbilde

4) 3 testa atbilde

5) poga “*Tālak*”

**c)**

1) Uzraksts “*Rezultāti, pareizo atbilžu skaits, procenti,*

*Atzīmes izvade…”*

2) Uzraksts  *“Atzīme”*

3) Uzraksts “*Procenti”*

4) Poga “*Beigt*”

# Priekšmetiskās jomas klašu sistēmas izstrāde

## Klašu diagramma

Klašu diagrammā ir realizēts vispārīgs klašu apraksts – klašu hierarhijas vispārējā struktūru, to sadarbība, atribūti, metodes, saskarnes un to savstarpējās attiecības, izmantojot UML valodu.



**2.1. att. Testēšanas sistēmas klašu diagramma UML valodā**

Balstoties uz izstrādāto sistēmas modeli var veikt sekojošu klašu iedalījumu (**2.1. att.**):

1. Klase **User** – ir atbildīga par darbu ar lietotājiem. Tā izveido jaunu lietotāju ar norādītiem datiem un nodrošina lietotāju pieslēgšanu sistēmai.

2. Klase **Admin** – nodrošina darbu ar lietotajiem un testiem …

3. Klase **Student** – nodrošina audzēkņa testa pildīšanas darbus …

4. Klase **Question** – ir atbildīga par darbu ar testa jautājumiem, to izvadi un ievadītas atbildes pārbaudi.

5. Klase **DistanceExamenator** – programmas darbību koordinējošā klase. Tā nodrošina programmas galvenās metodes izsaukumu un grafiskās saskarnes izveidi, tā arī …

## Klašu realizēšana java valodā

Izstrādātās klašu diagrammas (**2.1. att.**) 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. Klašu sistēma”, bet programmas kods, kas realizē šīs klašu sistēmas testēšanu (klase ar grafisko lietotāja saskarni) ir ievietots pielikumā “Pielikums B. testa programmas kods”.

class ***User***{

private String name;

private String login;

private String password;

private void User(String name, String login, String password){

}

private boolean enter(String login, String password){

return true;

}

}

class ***Admin*** extends User{

}

class ***Student*** extends User{

private int questionsCount;

private int rightAnswers;

private void getAnswer(Question question, String answer){

}

private void clear(){

}

}

class ***Question*** {

private String text;

private String answer;

private void Question(String text, String answer){

}

private void getText(){

}

private boolean isCorrect(String answer){

return true;

}

}

class ***DistanceExamenator*** extends Question{

private List users;

private List questions;

private User currentUser;

private void addUser(String name, String login, String password, String repeation){

}

private void findUser(String login, String password){

}

private void save(){

}

private void load(){

}

private void getResutls(){

}

public static void main(String[] args) {

}

}

# 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.x. att.**).

* Ielogošanas procesā ir iespējamas vairākas kļūdas – neaizpildīti lauki, neprecīzi ievadīti lietotāja dati.

**Login Form**

**Login:**

**Password:**

**Cancel**

**Login**

Click here to create a new account

**3.1. att. Testēšanas sistēmas ielogošanas forma**

**2. Reģistrācijas procesa apraksts:**

1) Galvenajā logā noklikšķiniet "*Create a new account*" apakšā

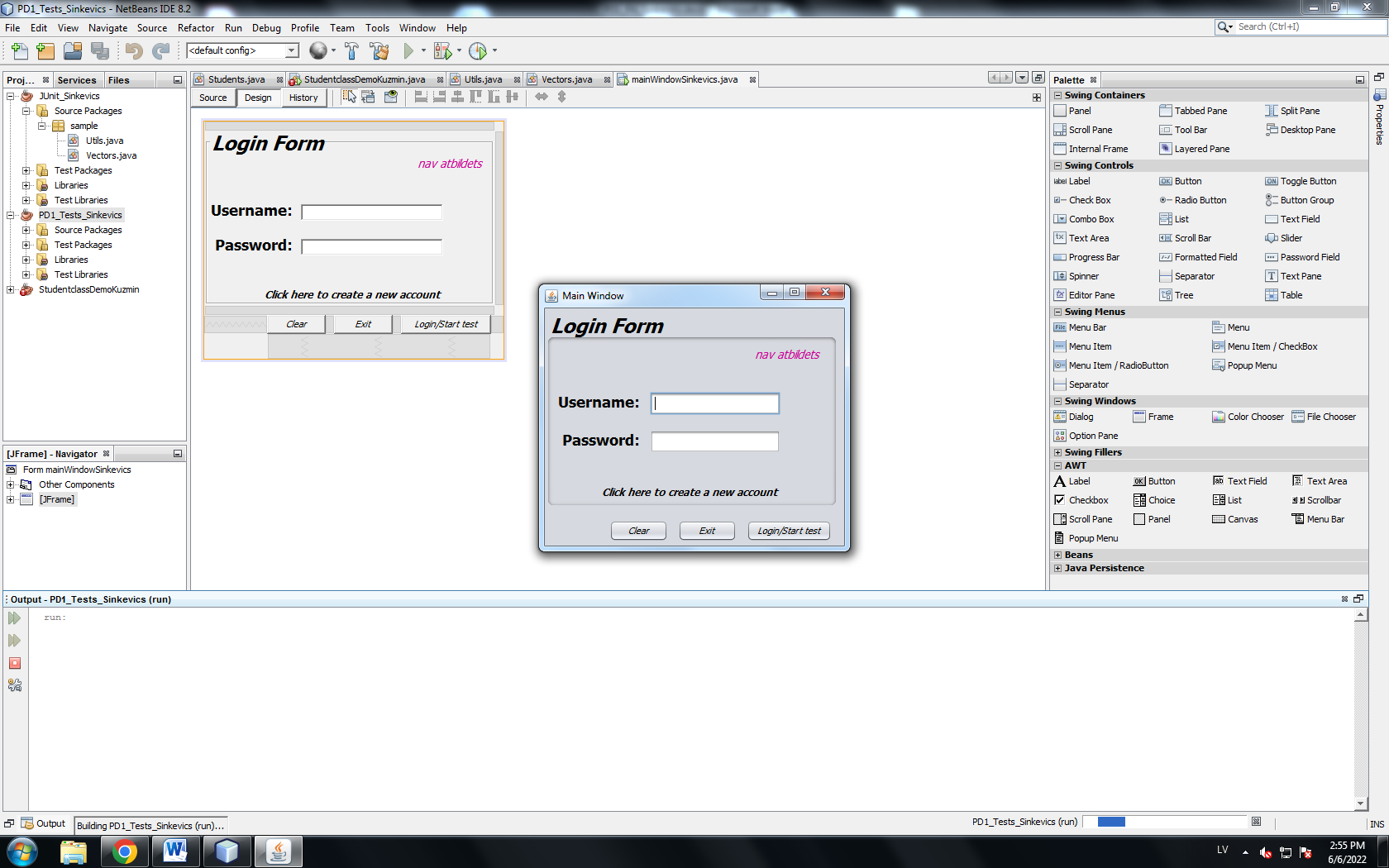
2) Aizpildiet visus nepieciešamos reģistrācijas laukus:

* teksta lauks “Vārds”
* teksta lauks “Uzvārds”
* teksta lauks “Lietotājvārds (Login)”
* teksta lauks ‘Parole”
* teksta lauks “Parole (atkārtoti)”

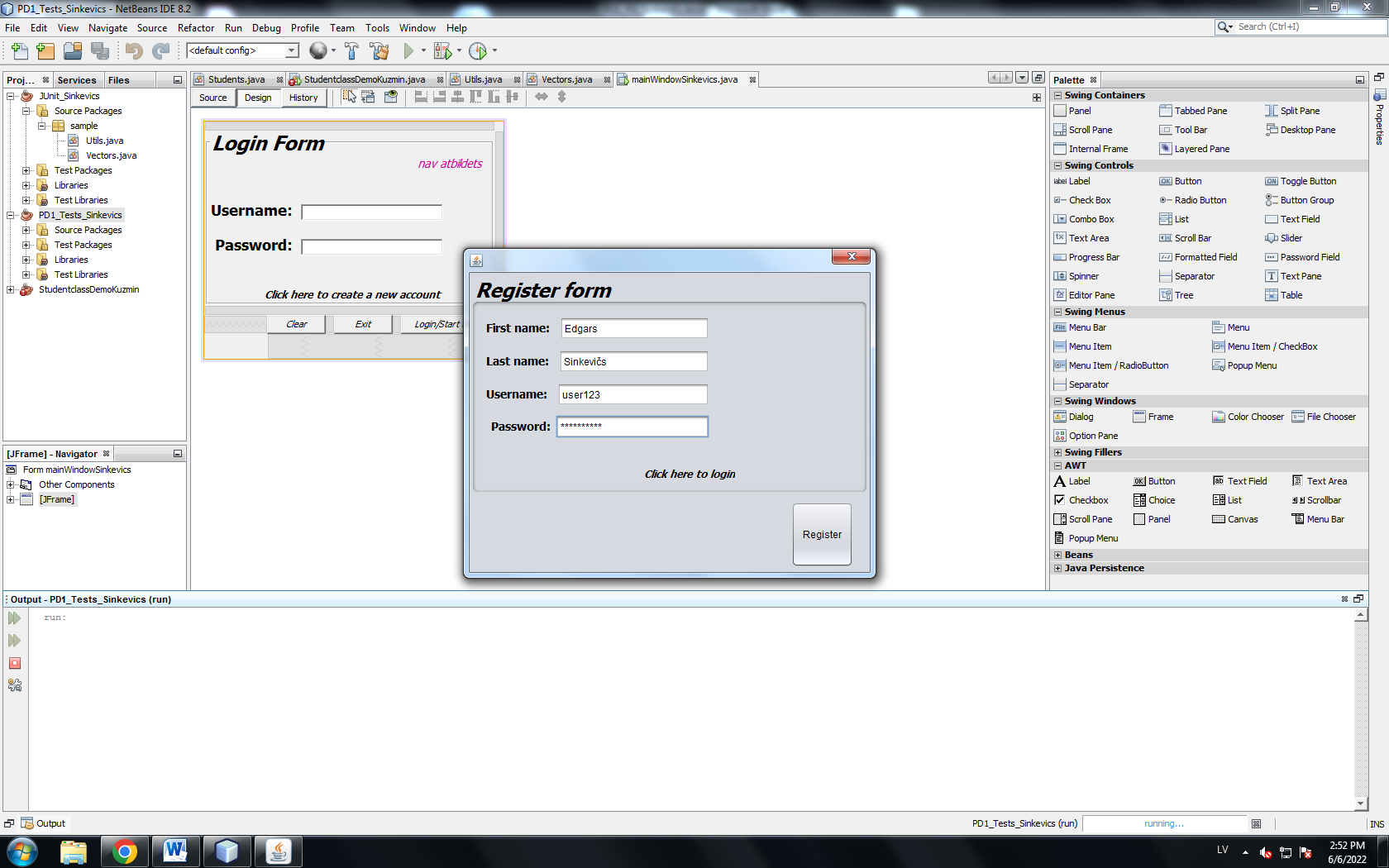
3) Pēc lauku aizpildīšanas noklikšķiniet uz pogas “*Reģistrēties*”.

4) Pēc reģistrācijas mēs ieejam galvenajā logā atbilstoši reģistrētajiem datiem.

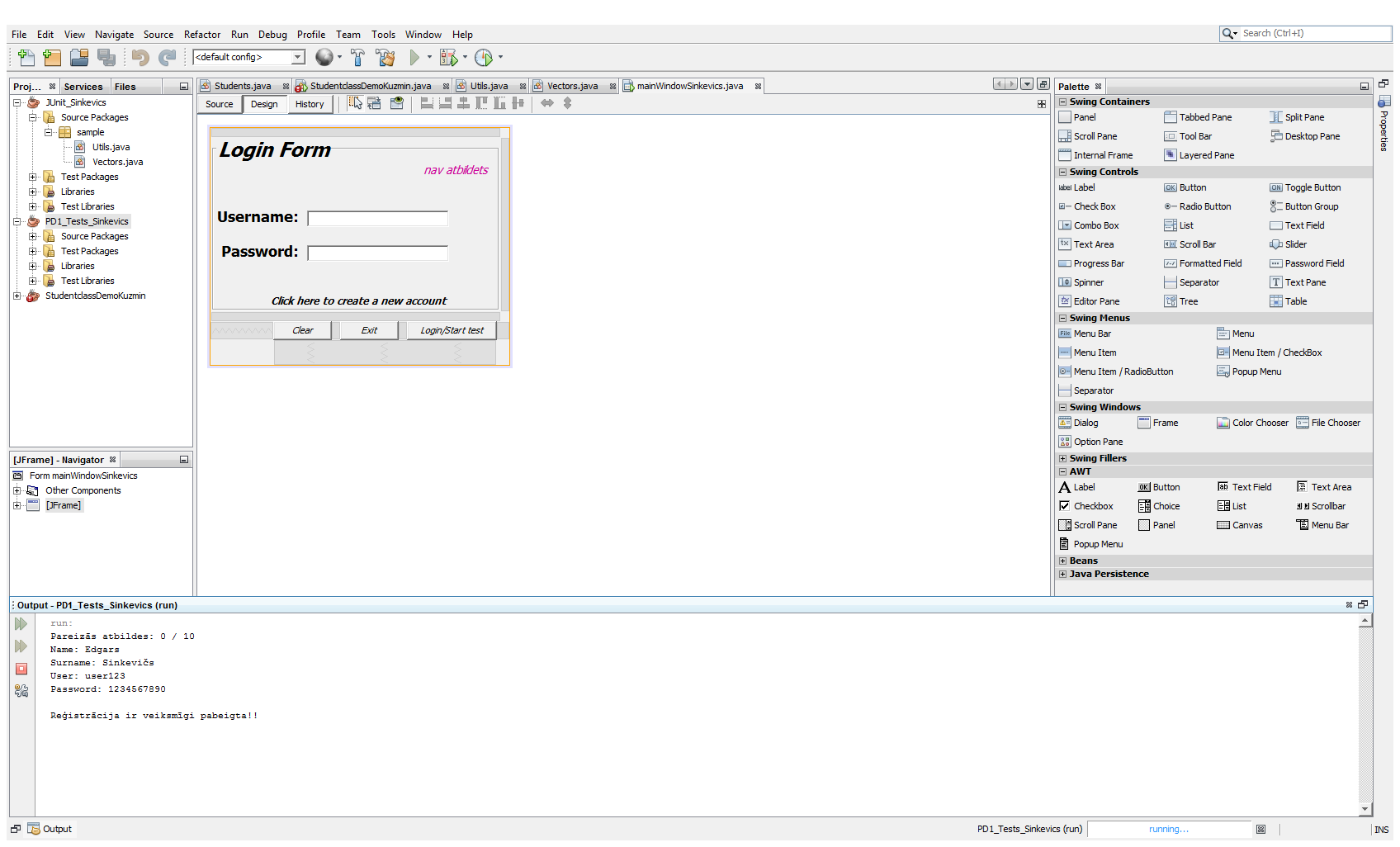
## Testēšanas rezultāti



*(Galvenais logs ar loginu un iespēju reģistrēties)*



*(Reģistrācijas logs)*



*(Reģistrēto datu izvade)*

# Lietotāja rokasgrāmata

## Lietotāja instrukcija

1. Lietotajs palaiz programmu

2. Piesakas sistema, izmantojot savu lietotajvardu un paroli, ja lietotajam nav konta, tavins tiek registrets.

2.1 Registracijas loga lietotajs aizpilda savus datus un noklikskina uz pogas Registreties.

3. Galvenaja izvelne lietotajs var ievadit 3 logus, kas atrodas izvelne.

3.1 Tests loga lietotajs var sakt izpildit testa uzdevumu.

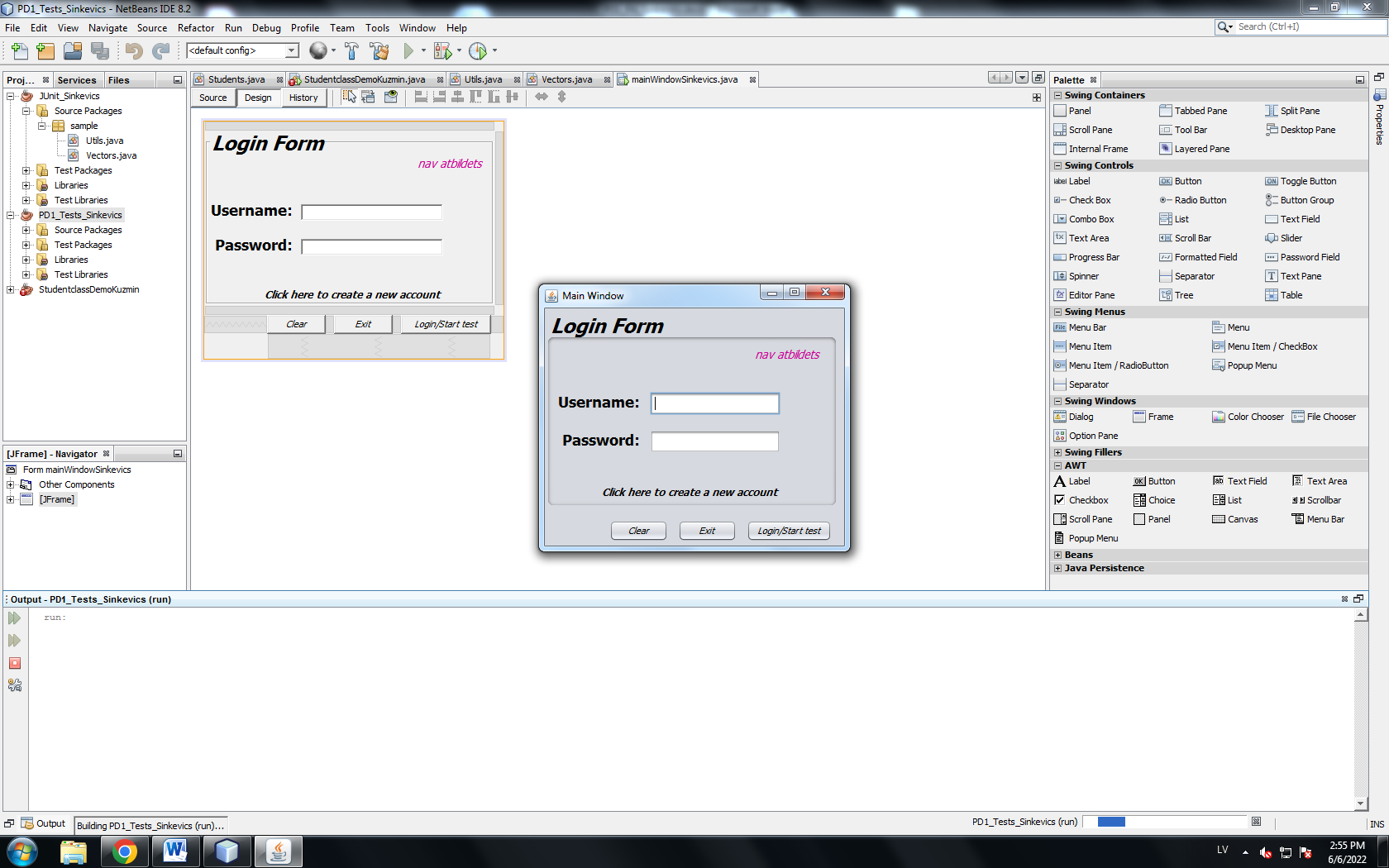
3.2 Mark loga varat redzet sava pedeja testa vertejumu.

3.3 Visus programmas lietotajus var apskatit Users izvelne.

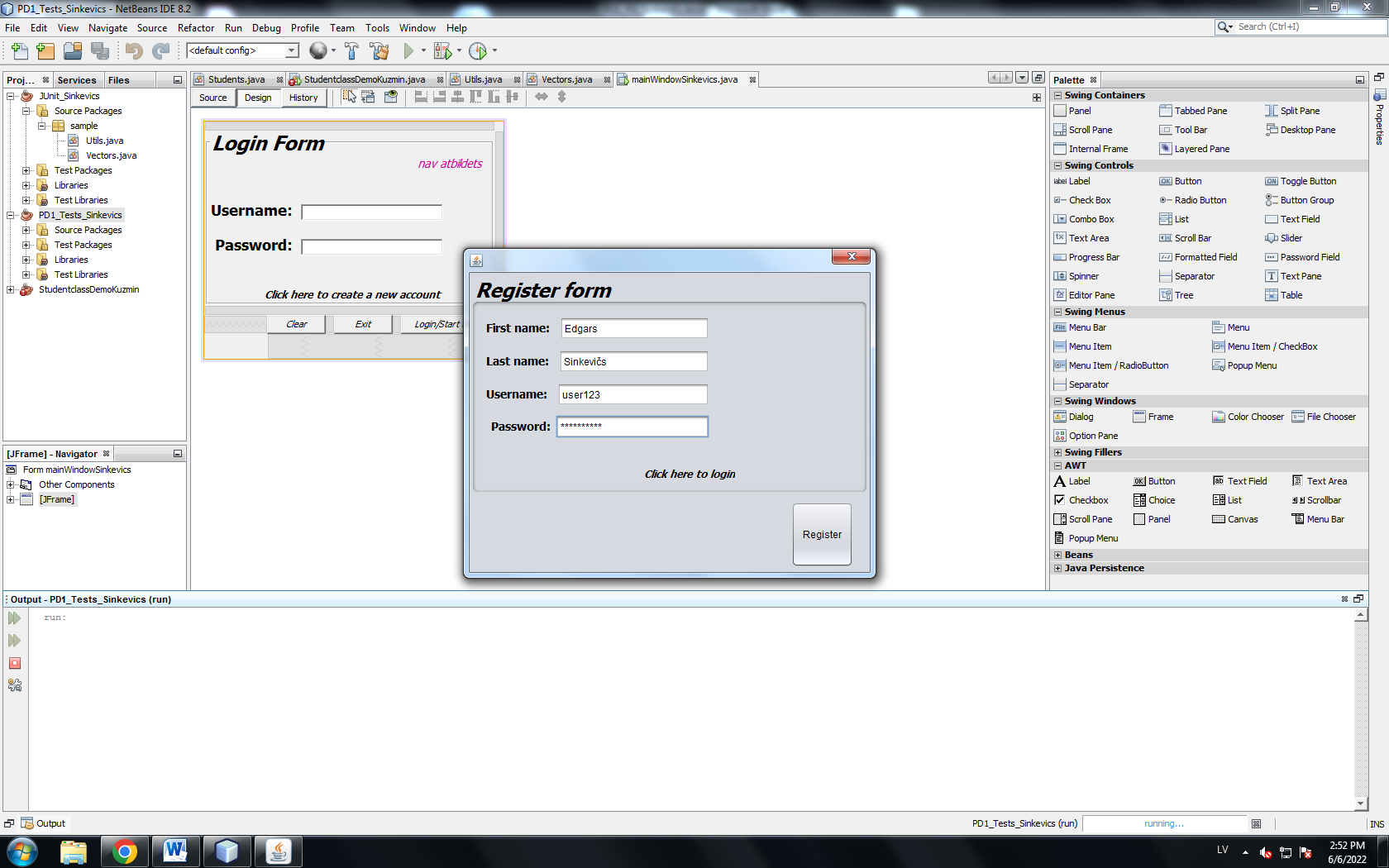
4. Izmantojot pogu LogOut, varat atteikties no sistemas.

## Palīdzības sistēma (help)

Padoms nereģistrētiem lietotājiem, ja viņiem nav konta, tad, sekojot ieteikumam, lietotājs varēs reģistrēties.



Padoms lietotājiem, kuri nejauši iekļuvuši reģistrācijas logā, un viņiem nav jāreģistrējas. Atgriežas pieteikšanās izvēlnē.



## Sistēmas ziņojumi

# Secinājumi

Projekta darbā es iemācijos aprakstīt testēšanas programmu, kura izpilda sekojošas funkcijas:

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

# Izmantotās literatūras (informācijas avotu) sarakst

# <http://e.daugvt.lv/>

# <https://morioh.com/p/f52f09775739>

# Pielikums a. klašu sistēma

# Pielikums b. testa programmas kods