**Rīgas Valsts tehnikums**

**Projekts**

Tēma: “Datorkluba administrācijas aplikācija”

Izpildīja: DP2-2 grupas studenti

Artem Bushev un Valnetīns Kaspers

Rīga, 2022 g.

Saturs:

[Projekta uzdevums 3](#_Toc103469769)

[Projekta tēma 3](#_Toc103469770)

[Projekta mērķis 3](#_Toc103469771)

[Uzdevumi 3](#_Toc103469772)

[Programmas prasības 3](#_Toc103469773)

[Priekšmeta jomas analīze 4](#_Toc103469774)

[Priekšmeta jomas apraksts 4](#_Toc103469775)

[Objektu apraksts 5](#_Toc103469776)

[Projektēšana 6](#_Toc103469777)

[Programmas struktūras projektēšana 6](#_Toc103469778)

[Metodes apraksts 10](#_Toc103469779)

[Programmu algoritmu projektēšana 11](#_Toc103469780)

[Kodēšana 13](#_Toc103469781)

[Galveno mainīgo apraksts 13](#_Toc103469782)

[Metodes apraksts 14](#_Toc103469783)

[Failu apraksts 15](#_Toc103469784)

[Testēšana 16](#_Toc103469785)

[Tests 1. Datu ierakstīšana datu bāzē 16](#_Toc103469786)

[Tests 2. Datu dzēšana 16](#_Toc103469787)

[Tests 3. Apskatīšana un filtrēšana 16](#_Toc103469788)

[Programmas darbības rezultāti 17](#_Toc103469789)

[Secinājumi 23](#_Toc103469790)

[Literatūras avoti 24](#_Toc103469791)

Projekta uzdevums

# Projekta tēma

Datorklubs

# Projekta mērķis

Nodrošināt efektīvu un ērtu datorklūba administratora darbaspēju ar atrūmu.

# Uzdevumi

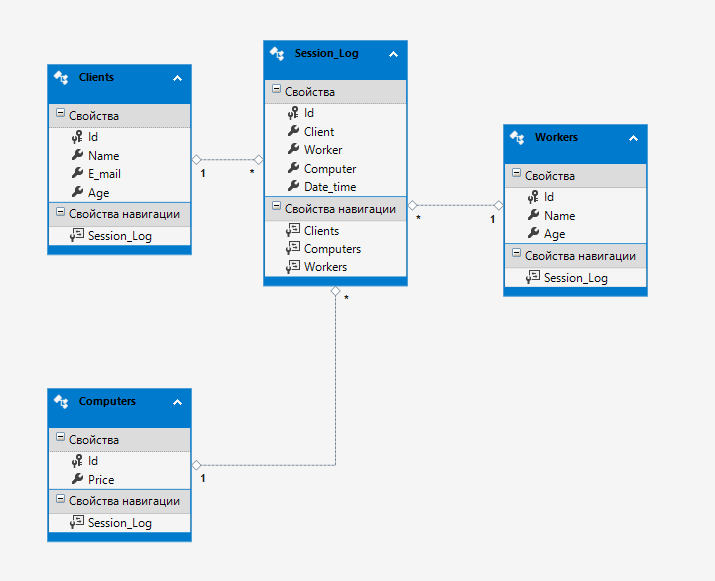
1. Jomas izvelešana
2. Objektu izveidošana
3. Metožu izstrāde
4. Izveidot programmas strukturu
5. Izveidot lietotāja intefeisu
6. Izstrādat algorītmus
7. Uzrakstīt kodu
8. Datu bazes izveidošana
9. Testēšana
10. Uzrakstīt atskaite

# Programmas prasības

1. Stradašana bez kļudam
2. Erts un saprotams interfeis
3. Interfeis angļu valoda

Priekšmeta jomas analīze

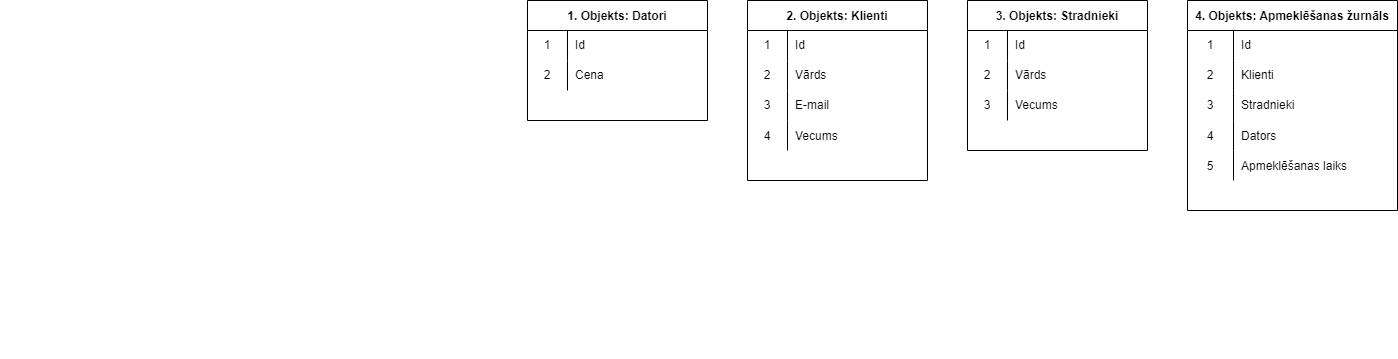
# Priekšmeta jomas apraksts

Programma bus izveidota datorkluba aministratorām, lai viņš varētu apskatīt visus datorus, reģistrēt, dzēst vai apskatīt lietotājus un lietotājus vai darbiniekus no savēja datora. Ierobežojums ir, datu filtrēšana jau ir iebuvēta programmā un viņu nevār izveleties, jā lietotajs ierakstīts apmeklējumas žurnala, tad viņu nevar izdzest.

1.att Tabulu attiecības

Šaja attēla var redzet datorkluba datu bāzes struktūru. Tris tabulas, Klienti, Datori un Stradnieki ir savienoti ar tabulu “Apmeklēšanas žurnals”. Tabulā Klienti P.K. ir Id, tapat kā tabulam Datori un Stradnieki. Tabulā “Apmeklēšanas žurnals” P.K. ir Id un F.K. ir Client, Worker un Computer.

# Objektu apraksts

Priekšmēta joma: Datorklūbs.

2.Att. Objektu apraksts

Objektam “Datori” ir Datora cena un Datora ID (identifikacijas numurs), Otrais objekts “Klienti” tur ir Vards, Vecums, Id un E-mail. Trešajs objekts “Stradnieki” tur ir Vards, Vecums un Id. Ceturtājs objekts ir “Apmeklēšanas žurnals” tur ir informācija par datokluba apmeklēšanu.

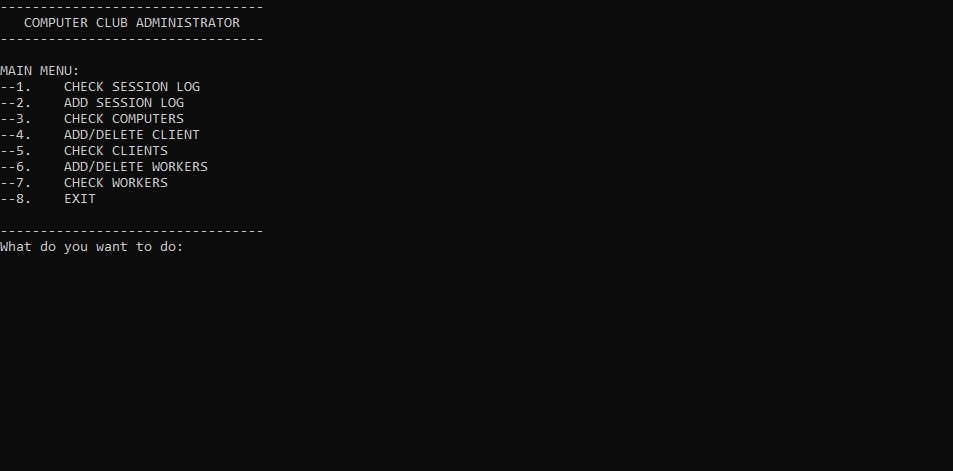
Projektēšana

# Programmas struktūras projektēšana

Programmas struktūra satur 8 moduļus:

1. Check session log
2. Add session log
3. Check Computers
4. Add/Delete client
5. Check clients
6. Add/Delete workers
7. Check workers
8. Exit
   * 1. Main menu

Formā ir šādas izvēlnes:

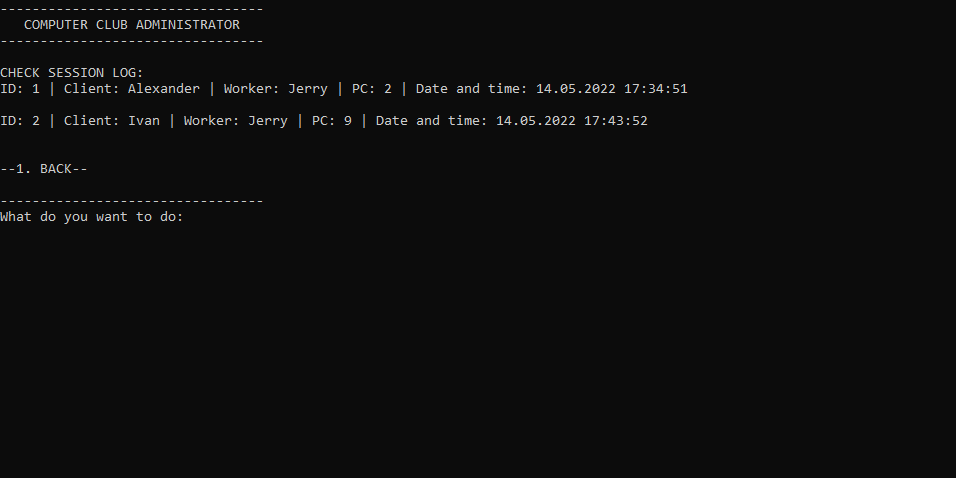
* + - 1. Check session log – parbaudīt apmeklēšanas žurnalu
      2. Add session log – pievienot jaunu apmeklējumu
      3. Check computers – apskatīt visus datorus
      4. Add/Delete client – pievienot vai dzēst klientu
      5. Check client – apskatīt klientus
      6. Add/Delete workers - pievienot vai dzēst stradniekus
      7. Check workers - apskatīt stradniekus
      8. Exit – Iziet no programmas

3.Att. Main Menu

* + 1. Check session log

Formā ir šādas izvēlnes:

1. Back – iziet uz Main Menu



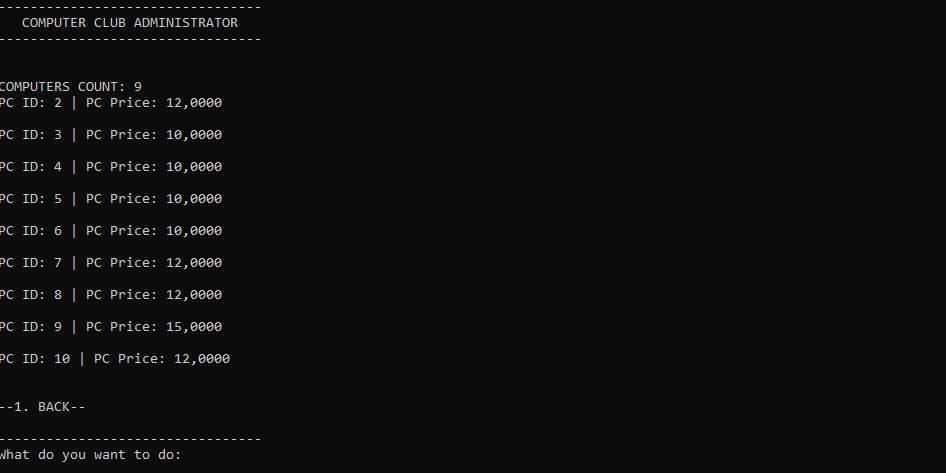
4.Att. Žurnala apskate

Attēla var redzēt, ka darbojas žurnals. Jus varat redzēt visas apmeklēšanas.

* + 1. Check Computer

Formā ir šādas izvēlnes:

* + - 1. Back – iziet uz Main Menu

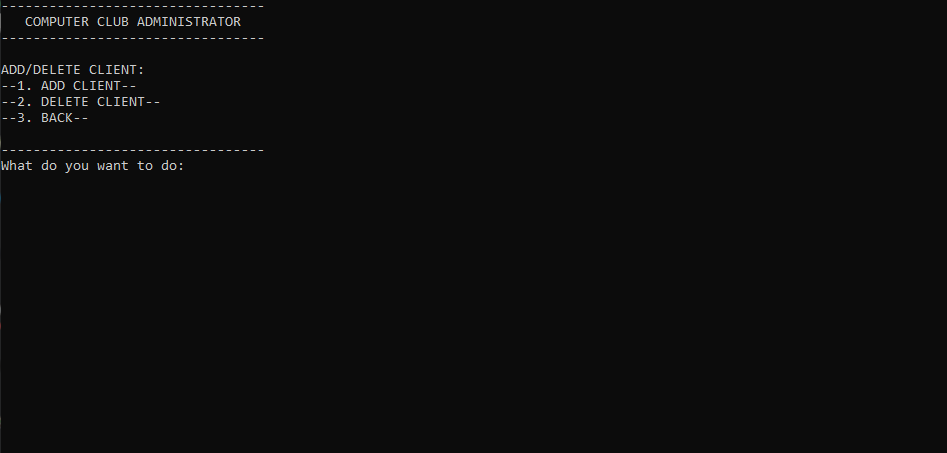


5.Att. Datora apskate

Attēla var redzēt, datora cenu uz 1 stundu un datora ID.

* + 1. Add/Delete client

Formā ir šādas izvēlnes:

* + - 1. Add client – pievienots jauno klientu
      2. Delete client – dzēst klientu no saraksta
      3. Back - iziet uz Main Menu

6.Att. Klienta pievienošana un dzēšana

Attēla var redzēt klienta pievienošanas un dzēšanas lpp.

* + 1. Check clients

Formā ir šādas izvēlnes:

* + - 1. Back - iziet uz Main Menu

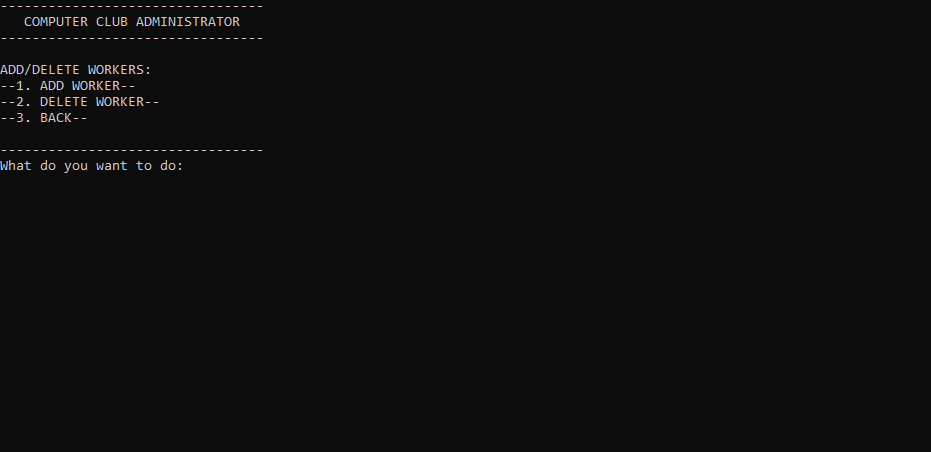
7.Att. Klienta apskate

Attēla var redzēt klienta apskatīšanas lpp.

* + 1. Add/Delete workers

Formā ir šādas izvēlnes:

* + - 1. Add workers - pievienots jauno strādnieku
      2. Delete client – dzēst stradnieku no saraksta
      3. Back - iziet uz Main Menu

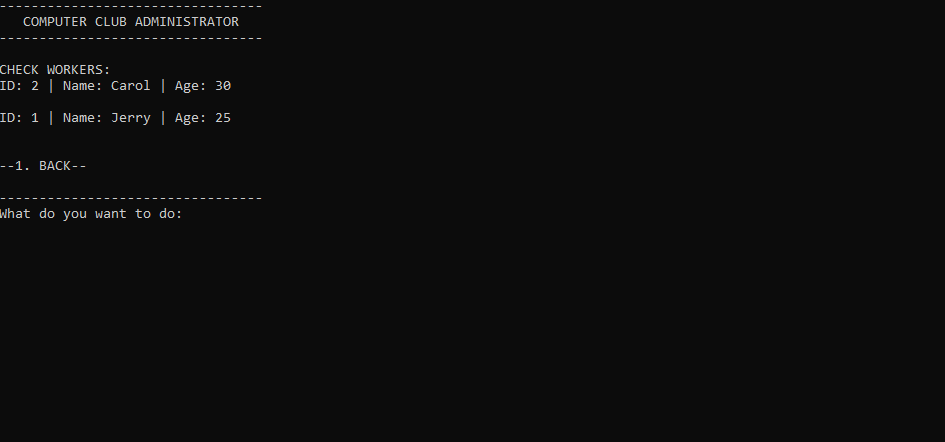


8.Att. Stradnieka pievienošana un dzēšana

Attēla var redzēt stradnieka pievienošanas un dzēšanas lpp.

* + 1. Check workers

Formā ir šādas izvēlnes:

* + - 1. Back - iziet uz Main Menu

9.Att. Stradnieka apskatīšana

Attēla var redzēt stradnieka apskatīšanas lpp.

# Metodes apraksts

Rakstot programmu, tiek izmantotas void tipa metodes. Tas nozīmē, ka metodes neatgriež rezultātu un nesatur return paziņojumu. Grafiskā interfeisa izstrādei tika izmantotas Console.WriteLine.

* + 1. Datu pievienošanas metodes.

1. void clientAdd();
2. void workerAdd();
3. void sessionlogAdd();
   * 1. Datu apskatīšanas metodes.
4. void workerOut();
5. void computerOut();
6. void clientOut();
7. void sessionlogOut();
   * 1. Datu dzēšanas metodes
8. void clientDelete();
9. void workerDelete();
   * 1. Interfeisa izvadīšanas metodes.
10. void clientEdit();
11. void clientCheck();
12. void computerCheck();
13. void sessionlogCheck();
14. void workersEdit();
15. void workersCheck();
16. void mainMenuStart();
17. void bannerStartup();

# Programmu algoritmu projektēšana

3.3.1. Datu pievienošanas metodes algoritms

Datu pievienošanas metodes algoritms programmā tiek izmantots metodēs void: void clientAdd(); void workerAdd(); void sessionlogAdd();. Šīs metodes algoritms darbojas šādi:

1. Lietotājs ievada nepieciešamos datus, kas galu galā tiek piešķirti mainīgajiem.
2. Mainīgo vērtības tiek nodotas mainīgajiem tabulās
3. Izmaiņas tabulās tiek saglabātas
4. Konsole parāda paziņojumu, ka pievienošana bija veiksmīga.
   * 1. Datu dzēšanas metodes algoritms

Datu dzēšanas metodes algoritms programmā tiek izmantots metodēs void: void clientDelete(); void workerDelete(); Šīs metodes algoritms darbojas šādi:

1. Ievadiet klienta ID
2. Mainīgais mainīgais ņem ievadīto vērtību un nekavējoties tiek pārveidots par Int32 datu tipu
3. linq vaicājums, kurā no datu bāzes tabulas tiek apskatīti visi klienti/darbinieki un atlasīti tie klienti/darbinieki, kuru ID ir vienāds ar ievadīto numuru
4. Ja pieprasījums nav tukšs, tad programma pieprasa darbības apstiprinājumu un dzēš darbinieka klientu, bet ja pieprasījums ir tukšs, programma neko neizdzēsīs.
   * 1. Datu apskatīšanas metodes

Datu apskatīšanas metodes algoritms programmā tiek izmantots metodēs void: void workerOut(); void computerOut(); void clientOut(); void sessionlogOut(); Šīs metodes algoritms darbojas šādi:

1. Saraksta izveide
2. Saraksts ir aizpildīts ar datiem no tabulas
3. Foreach cikls pārbauda visas sarakstā esošās vērtības
4. Datu izvade no saraksta

Kodēšana

# Galveno mainīgo apraksts

|  |  |  |
| --- | --- | --- |
| **Galvenie** **mainīgie** **un** **tas** **apraksts** | | |
| № | **Mainīgais** | **Apraksts** |
| 1, | string Name | *Klienta vai stradnieka vārds* |
| 2, | string E\_mail | *Klienta e-mail* |
| 3, | string Age | *Klienta vai stradnieka vecums* |

1.tabula mainīgo apraksts

Šaja tabulā ir redzami tris galvenie mainīgie string datu tipa, Name, E\_mail un Age.

1. string Name – atbilst klienta vai stradnieka vārdam.
2. string E\_mail – atbilst klienta E\_mail.
3. string Age – atbilst klienta vai stradnieka vēcumam

# Metodes apraksts

|  |  |  |  |
| --- | --- | --- | --- |
| Metodes nosaukums un merķis | | | |
|
| № | Tips | Nosaukums | Merķis |
|
| 1. Intefeisa metodes | | | |
|
| 1 | void | bannerStartup(); | Programmas nosaukuma attēlojums |
|
| 2 | void | mainMenuStart(); | Menu attēlojums |
|
| 3 | void | clientEdit(); | Klienta rediģešanas attēlojums |
|
| 4 | void | clientCheck(); | Klienta apskatīšanas attēlojums |
|
| 5 | void | computerCheck(); | Datora apskatīšanas attēlojums |
|
| 6 | void | workersEdit(); | Stradnieka rediģešanas attēlojums |
|
| 7 | void | workersCheck(); | Stradnieka apskatīšanas attēlojums |
|
| 8 | void | sessionlogCheck(); | Žurnala apskatīšana |
|
| 2. Galvenie metodi | | | |
|
| 1 | void | clientAdd(); | Klienta pievienošana |
|
| 2 | void | workerAdd(); | Stradnieka pievienošana |
|
| 3 | void | sessionlogAdd(); | Apmeklējuma pievienošana |
|
| 4 | void | workerOut(); | Stradnieka saraksta izvade |
|
| 5 | void | computerOut(); | Datora saraksta izvade |
|
| 6 | void | clientOut(); | Klienta saraksta izvade |
|
| 7 | void | sessionlogOut(); | Žurnala saraksta izvade |
|
| 8 | void | clientDelete(); | Klienta dzēšana |
|
| 9 | void | workerDelete(); | Stradnieka dzēšana |
|

2.tabula Metodes nosaukums un merķis

Šaja tabulā ir redzami 8 interfeisa metodes un 9 galvenas metodes.

Interfeisa metodes:

1. void bannerStartup();
2. void mainMenuStart();
3. void clientEdit();
4. void clientCheck();
5. void computerCheck();
6. void workersEdit();
7. void workersCheck();
8. void sessionlogCheck();

Galvenie metodi:

1. void clientAdd();
2. void workerAdd();
3. void sessionlogAdd();
4. void workerOut();
5. void computerOut();
6. void clientOut();
7. void sessionlogOut();
8. void clientDelete();
9. void workerDelete();

# Failu apraksts

|  |  |  |
| --- | --- | --- |
| Faili un struktūra | | |
|
| Tabula | Mainigais | Apraksts |
|
| Klients | int Id | Klienta ID |
|
| nvarchar(50) Name | Klienta vārds |
|
| nvarchar(50) E\_mail | Klienta e-mail |
|
| int Age | Klienta vecums |
|
| Dators | int Id | Datora ID |
|
| money Price | Maksa par stundu (eiro) |
|
| Stradnieks | int Id | Stradnieka ID |
|
| int Age | Stradnieka vecums |
|
| nvarchar(50) Name | Stradnieka vārds |
|
| Apmeklēšanas žurnals | int Id | Žurnala ID |
|
| int Client | Klienta ID |
|
| int Worker | Stradnieka ID |
|
| int Computer | Datora ID |
|
| datetime Date\_time | Ierakstīšanas datums un laiks |
|

3.tabula Faili un struktūra

Šaja tabulā ir redzama visa struktūra ar failiem:

Tabulai “Klients” ir 4 mainigie - int Id, nvarchar(50) Name, nvarchar(50) E\_mail, int Age:

1. int Id – atbild par klienta ID
2. nvarchar(50) Name – atbild par klienta vārdu
3. nvarchar(50) E\_mail – atbild par klienta E\_mail
4. int Age – atbild par klienta vecumu

Tabulai “Dators” ir 2 mainigie - int Id, money Price:

1. int Id – atbild par datora ID
2. money Price – atbild par datora cenu

Tabulai “Stradnieks” ir 3 mainigie - int Id, int Age, nvarchar(50) Name:

1. int Id – atbild par stradnieka Id
2. int Age – atbild par stradnieka vecumu
3. nvarchar(50) Name – atbild par stradnieka vārdu

Tabulai “Apmeklēšanas žurnals” ir 5 mainigie - int Id, int Client, int Worker, int Computer, datetime Date\_time:

1. int Id – atbild par apmeklēšanas Id
2. int Client – atbild par klienta Id
3. int Worker – atbild par stradnieka Id
4. int Computer – atbild par datora Id
5. datetime Date\_time – atbild par apmeklēšanas laiku

Testēšana

# Tests 1. Datu ierakstīšana datu bāzē

Pārbaudot datu pievienošanu, bija problēmas. Dati netika saglabāti datu bāzē. Problema tika risināta ar datu bāzes saites atjaunošanu.

# Tests 2. Datu dzēšana

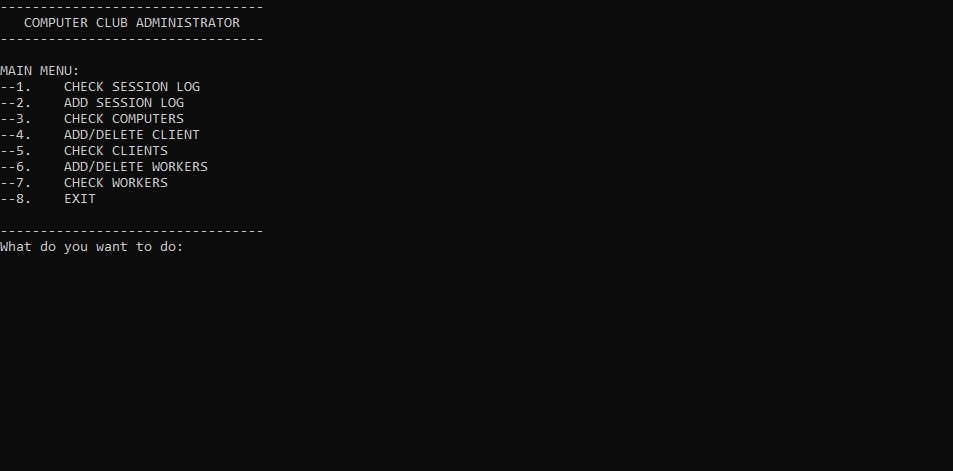
Ievadot klienta vai darbinieka ID, nebija problēmu atrast nepieciešamo lauku. Arī pēc izņemšanas problēmas ar datu bāzi nebija.

# Tests 3. Apskatīšana un filtrēšana

Apskatot darbinieku un klientu sarakstu, problēmas ar informācijas izvadīšanu no datu bāzes, piemēram, nepareiza datu tipa konvertēšana un nepareiza datu izvade, netika konstatētas. Alfabētiskā filtrēšana ir veiksmīga

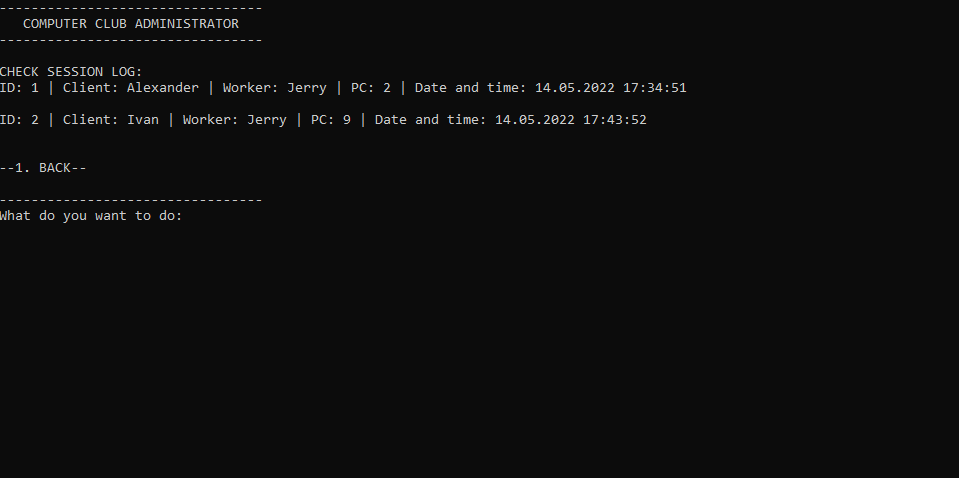
Programmas darbības rezultāti

Kad tiek palaists \*.exe fails, tiek atvērta galvenā izvēlne informācijas apstrādes izvēlei:



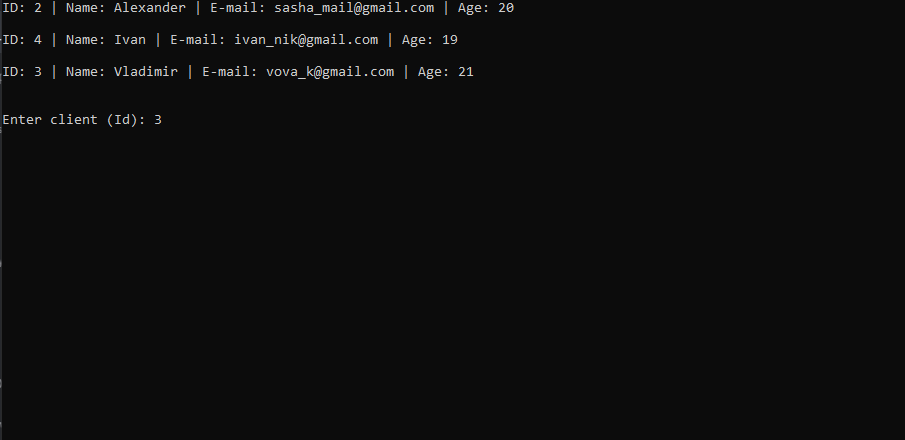
10.Att. Main menu

Ievadot "1", lietotājam ir iespēja apskatīt visus apmeklējumus.



11.Att. Sesssion log apskatē

Ievadot "2", lietotājam ir iespēja izveidot un uzrakstīt jaunu apmeklējumu



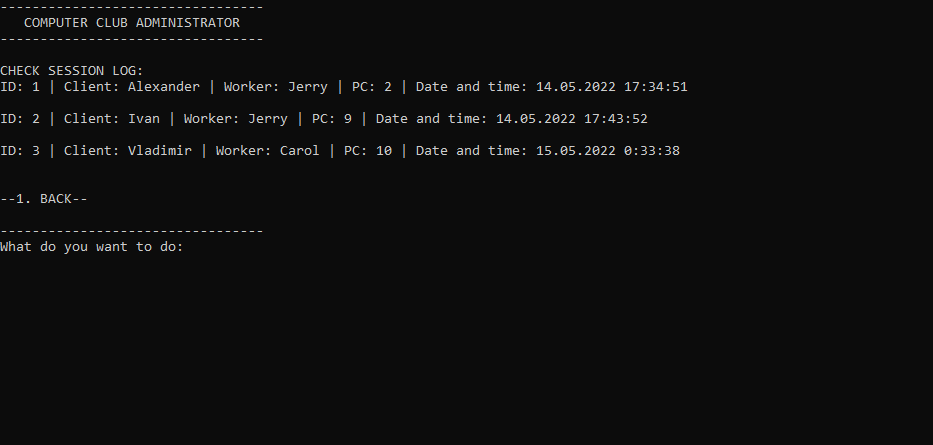
12.Att. Ievadam klienta ID



13.Att. Ievadam stradnieka ID

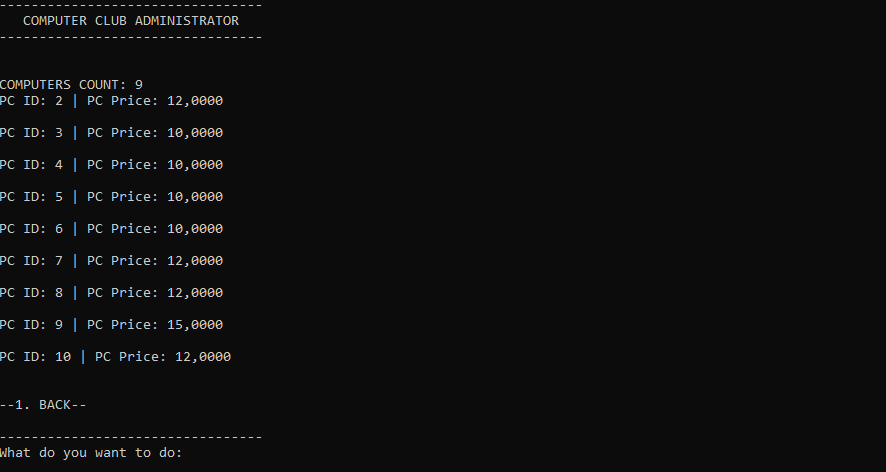


14.Att. Ievadam datora ID



15.Att. Jauns apmeklējums izvediots (klienta/sradnieka pievienošsana strāda tapat, kā šeit)

Ievadot "3", lietotājam ir iespēja apskatīt visus datorus

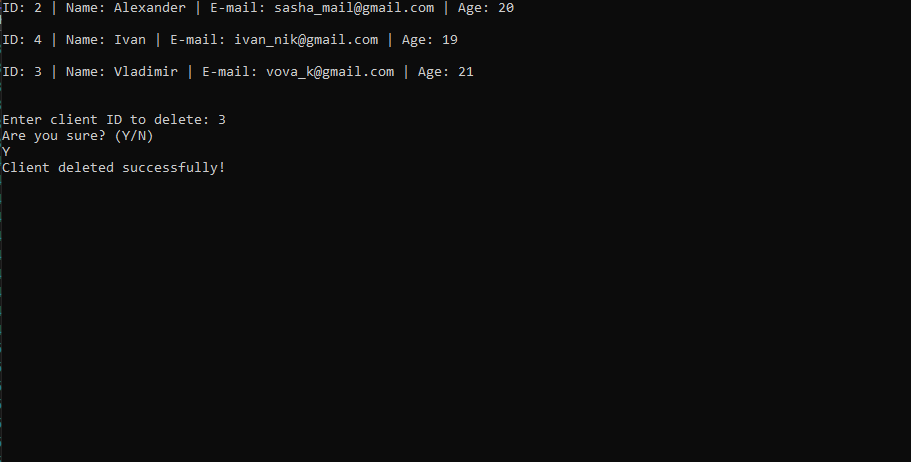


16.Att. Datora apksatē

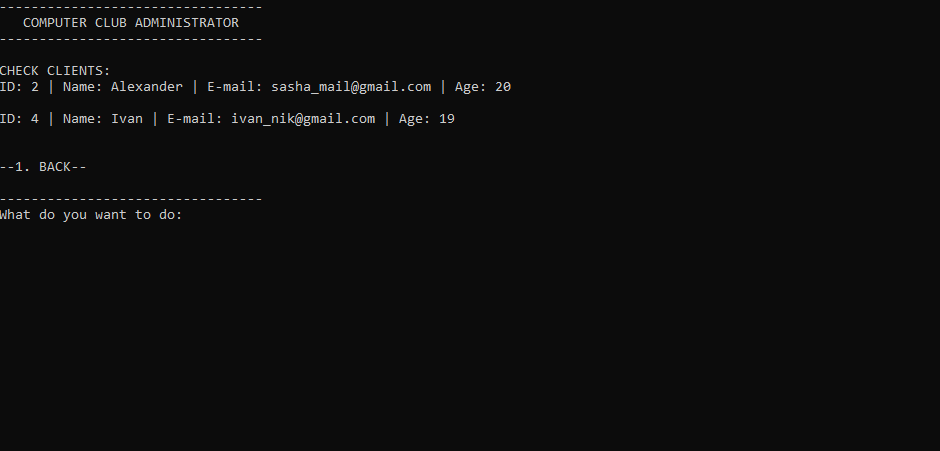
Ievadot "4", lietotājam ir iespēja pievienot jaunu vai izdzēst klientu



17.Att. Klienu dzešana

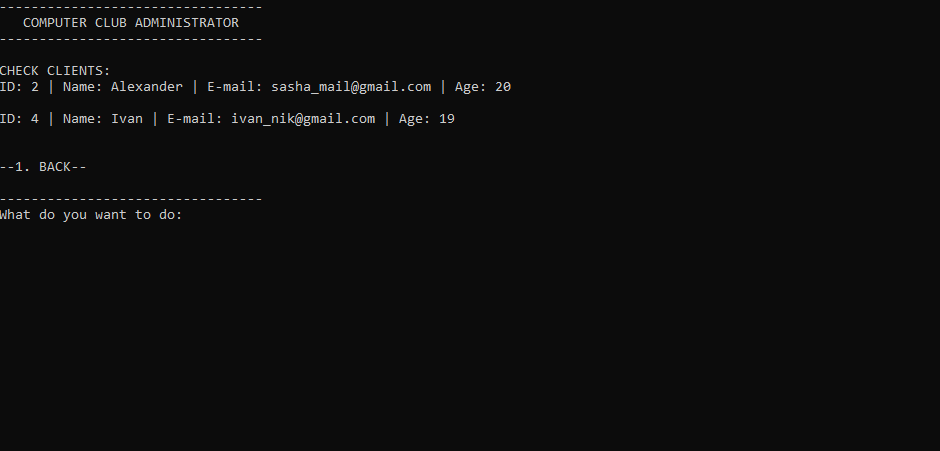


18.Att. Izdzesām klientu



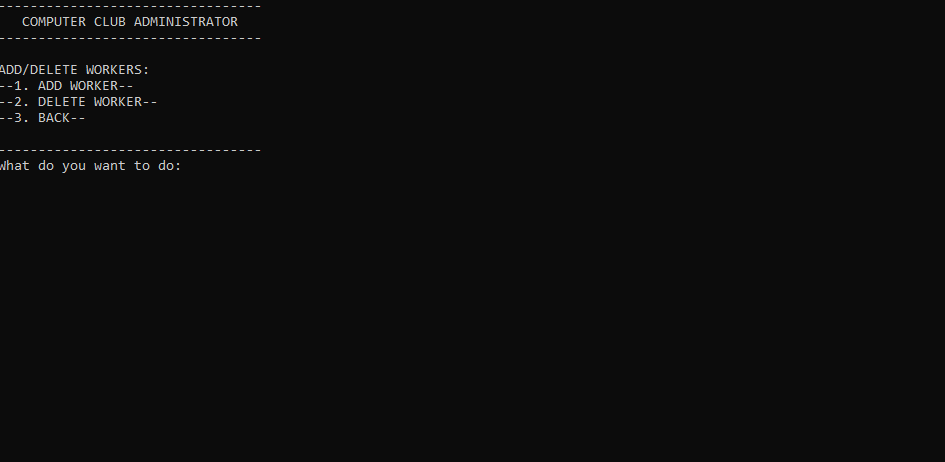
19Att. Klients izdzests (izdzēšana stradniekime strāda tapat, kā šeit)

Ievadot "5", lietotājam ir iespēja apskatīt klientus



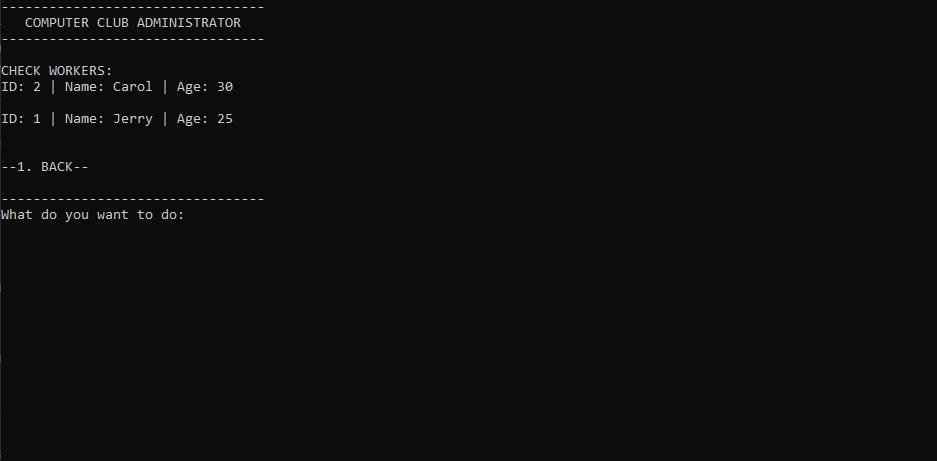
20.Att. Klientu apskatē

Ievadot "6", lietotājam ir iespēja pievienot jaunu vai izdzēst stradnieku



21.Att. Darbinieku pieveinošana vai dzešana

Ievadot "7", lietotājam ir iespēja parbaudīt stradniekus



22.Att. Stradnieku apskate

Ievadot "8", lietotājam ir iespēja iziet no datora kluba (no programmas)

Secinājumi

Šī darba gaitā izdevās izveidot konsoles aplikāciju datorkluba administrēšanai. Projektā tika realizēta datubāze, pievienojot un dzēšot datus, kā arī apskatot tabulu ar informāciju no klientiem un visiem kluba darbiniekiem. Veidojot aplikāciju, radās problēmas ar datu bāzes savienošanu, kā arī algoritma izveidi datu dzēšanai un pievienošanai tabulām. Šis darbs kopumā aizņēma aptuveni 3 nedēļas. Lietojumprogrammai vēlams pievienot arī metodes izvēles pārbaudi. Koda izpēte un labošana lika man pievērsties oficiālajai Microsoft dokumentācijai un tiešsaistes resursam metanit.com.

Literatūras avoti

Internets:

1. Microsoft dokumentācija
2. Metanit.com