SVEUČILIŠTE U SPLITU SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij Informacijske tehnologije

Josip Pavić

ZAVRŠNI RAD

IZRADA DESKTOP APLIKACIJE KARTAŠKE IGRE BRISKOLA KORISŠTENJEM FORMI

Split, lipanj 2019

SVEUČILIŠTE U SPLITU SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij Informacijske tehnologije

Josip Pavić

ZAVRŠNI RAD

IZRADA DESKTOP APLIKACIJE KARTAŠKE IGRE BRISKOLA KORISŠTENJEM FORMI

Split, lipanj 2019

SVEUČILIŠTE U SPLITU SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Specijalistički diplomski stručni studij Informacijske tehnologije

Predmet: Programiranje u C#

Z A V R Š N I R A D

Kandidat: Josip Pavić

Naslov rada: Naslov završnog rada

Mentor: Marina Rodić, predavač

Split, lipanj 2019

Sadržaj:

Sažetak......................................................................................................1

1.Uvod.........................................................................................................

2. Tehnologije..............................................................................................

2.1 C#

**2.2 Entity Framework**

2.2 JSON

2.3 TCP

2.4 **ASP.NET**

3.Aplikacija.................................................................................................

4.Zaključak.................................................................................................

5.Literatura.................................................................................................

6.Popis slika................................................................................................

**SAŽETAK**

Cilj ovoga završnoga rada bio je izrada aplikacije u programskom jeziku C# koja će služiti u svrhu zabave korisnika.

Aplikacija je izrađena u C# programskom jeziku.

Opisane su tehnologije koje će se koristiti pri izradi aplikacije.

Projekt se sastoji od desktop aplikacije i web aplikacije.

Desktop i web aplikacija u pozadini koriste istu bazu podataka.

Baza podataka je izrađena u Miscorsoft SQL Menagement Studio 2017 , a nalazi se na azure serveru.

Desktop aplikacija omogućava korisniku da kreira pristupnik(server) ili da kao klijent pristupi nekom od postojećih pristupnika(servera).Sama igra se odvija u desktop aplikaciji.

Igra je realizirana tako da omogućava igru između dva korisnika,te ishod pohranjuje u bazu podataka.

Podaci o statusu korisnika se ažuriraju u bazu podataka,ćime je omogučeno da korisnik

ima pregled na listu igraća koja je trenutno prijavljena u aplikaciju.

Također korisnik ima pregled i svih servera koji su trenutno podignuti.

Web aplikacija omogućava aministratoru jednostavnije upravljanje bazom.jer ne zahtjeva instalaranu dekstop aplikaciju.Korisniku je također omogućen lakši pregled i uređivanje profila putem web aplikacije.

**2.TEHNOLOGIJE**

2.1 ASP .NET MVC

Tehnologija Microsoft .NET okvira koja se koristi za razvoj dinamičkih web stranica,interaktivnih web stranica,te web servisa.

ASP stranice se izvršavaju na serverskoj strani te generiraju kod u HTML,XML formatu koji se šalje desktop ili mobilnim preglednicima.

MVC kodna arhitektura,koja je razvijana prvenstveno za desktop aplikacije danas je popularnija na web platformama.

Osnovna ideja MVC arhitekture je razdvajanje koda na tri cjeline.

MVC(model,pregled,kontroler) je okvir koji se koristi za izradu web aplikacija otvorenoga koda, a baziran je na MVC obrascu.Potpuno je neovisan o platformi te je definiran na posebnom asembleru System.Web.Mvc.

ASP .NET MVC arhitektura se sastoji od tri međusobno zavisne komponente:

Model-u sloju modela je najčešće implementirana poslovna logika,tj.to je skup klasa koje opisuje podatke poslovne logike s kojom se radi.Model enkapsulira podatke koji se čuvaju u bazi podataka.

Pogled-omogućava prikaz podataka,i najčešće se koristi za prikaz podataka iz modela.

Upravitelj -veza između modela i pogleda,čita ulazne podatke od korisnika te ih prosljeđuje modelu.Nakon komunikacije s modelom Upravitelj odlučuje koji će se sadržaj prikazati krajnjem korisniku.

**MVC je baziran na kontrolorima.**

**ASP .NET MVC za čuvanje podataka koristi metode koje se nalaze u kontroloru.**

**ASP .NET MVC ima mogućnost korištenja i REST čime se metode kontrolera pokreću preko URL-a.**

**2.2Entity Framework**

**Entity framework je okvir objektno-relacijskog mapiranja(ORM) otvorenog koda za ADO.NET.**

**Od verzije 6 je odvojen od .NET okvira.**

**Entity okvir aplikacija se može pokrenuti na bilo kojem računalu koje ima instaliran .NET okvir.**

**Entity framework je postao jedna od vodećih tehnologija za pristup podacima**

**Entity framework omogućava jednostavnu sinkronizaciju modela s bazom podatakaEntity framework ima mogučnost da automatski generira kod ćime značajno štedi vrijeme programeru.**

**Posjeduje više načina da se napravi ORP mapiranje,tj. postoje tri različita development Workflowa.**

**1.Baza prva-generira sve potrebne objekte na osnovi modela baze podataka**

**2.Kod prvi-generira sve potrebne bazne objekte na osnovi koda**

**3.Model prvi-definira entitete i ????????? na osnovi koji će se generirati baza podataka i klase**

**JSON-je tekstualni format čija je namjena prijenos podataka u formatu koji je čitljiv i ljudima i strojevima.Ekstenzija koja se koristi za json datoteku je oblika .json.**

**Json zbog svoje jednostavnosti sve više istiskuje xml,koji koristi oznake zbog čega je teži za pisanje i čitanje. Prednost json-a je što za parsiranje koristi js parser.**

**Objekt u json formatu je oblika ključ:par a nalazi se unutar vitičastih zagrada.**

**JSON ne ovisi o programskom jeziku,a njegova najveća primjena je u web aplikacijama.**

**Json podražaje različite tipove podatak kao: tekst,bolean,broj,polja.**

Glavna razlika jsona u odnosu na java skript je sto json ne može sadržavati funkciju.

**Windows forme**

Windows forma je alat koji se koriste za izradu windows aplikacija.

Windows forme su učinkovit i jednostavan način koji omogućava korisniku komunikaciju sa programom.

Toolbox(alati) sadrži skup gotovih windows kontrola.

Toolbox sadrži kontrole koje se mogu dodati windows forms aplikacijama.

Dizajn prikazuje obrazac forme sa svim elementima koji se nalaze u formi.

Svojstva(properties) nam omogućavaju da podesimo postavke za željeni element,a dijele se na devet kategorija:izgled,ponašanje,podaci,pristupačnost,dizajn,fokus,predložak,stil prozora i različito.

Istraživač rješenja(Solution Explorer) koristi se za prikaz strukture projekta.Za izradu windows formi potrebno je unutar Visual Studia odabrati File->New projekt te se u izborniku odabire windows forma,upiše se željeno ime te se pritiskom na OK botun kreira nova windows form aplikacija.

**C#**

C# je u potpunosti objektno-orijentiran programski jezik,što znači da podržava koncešte nasljeđivanja ,enkapsulacije,polimorforizma.

C# je tipiziran jezik,što znači da se mora obaviti deklaracija svakog stvorenog objekta,a prevoditelj će prijaviti pogrešku ako objektu nije pridružen odgovarajući tip podataka.

Svi tipovi podataka imaju određenu i nepromjenjivu veličinu npr.tip short se preslikava u int16.

U C# vrši se podjela na dva tipa podataka: vrijednosne(bolean,char,integer,decimal,enum) i referentne(Klase,Sučelja,Delegati,Polja..).

Osnovna razlika je u načinu pristupa te što se vrijednosni tipovi pohranjuju na stogu,a referentni na kontroliranom heap-u.

Varijablama je potrebno pridružiti vrijednost ili obaviti inicijalizaciju prije samog korištenja.Program se izvršava u CLR,to omogućava ponovno korištenje klase koja je napisana u nekom od .NET programskih jezika.

C# upravlja memorijom koristeći svoj sustav za skupljanje otpada(Garbage Collector).

C# tekstualne datoteke sadrže ekstenzije .cs .

**TCP**

**Tcp se uglavnom koristi u svim aplikacijama koje zahtjevaju pouzdanu isporuku paketa između dva krajnja uređaja.Omogućava višestruko istrovremeno povezivanje klijenata na server.**

**Tcp za prijenos korisničkih podataka koristi IP.Ip protokol sadrži informacije o uređajima koji međusobno komuniciraju**

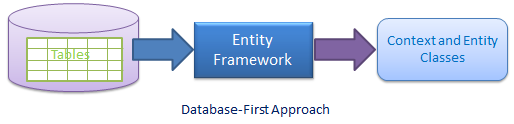
3.APLIKACIJA

**3.1 BAZA PODATAKA**

Nakon zavrsetka izrade same igre,bilo je potrebno osmisliti model i izgled baze podataka.Baza podataka se koristi za pohranjivanje registriranih korisnika,te za pohranjivanje konačnog ishoda igre.

Baza podataka sastoji se od 5 tablica.Relacije između tablica su prikazane na slici 3.1.

Za izradu baze podataka koristio je se baza prva pristup ,kod kojeg se prvo modelira baza podataka ,na temelju koje će entity razvojni okvir(Entity Framework) amutomatski generirati potrebne modele.Na slici 3.1.1 je prikazan baza prva model entity razvojnog okvira.

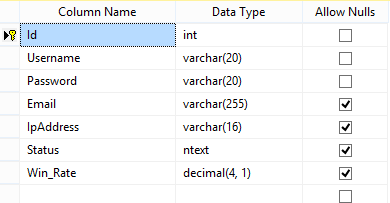
[](https://www.entityframeworktutorial.net/images/EF5/databasefirst.png)

Slika 3.1.1 EF baza prva schema

Baza podataka je izrađena pomoću Microsoft SQL Server Menagment Studio 2017 alata.

Nakon izrade baze podataka sljedeći korak je izrada pojedinih tablica .

Primjer izrađene tablice



Slika 3.1.2



Slika 3.1.3 EER model

Slika 3.1.3 prikazuje EER model baze podataka koja sadrži tablice sa odgovarajućim elementima.Na slici prikazane su i relacije između pojedinih tablica.

Relacije između tablica važne su za lakšu manipulaciju samim podacima,te pravilan rad baze podataka.

Podaci o svim korisnicima,korisničkim ulogama(role), igri(game), pohranjeni su u bazi podataka Players.mdf.Baza podataka je kreirana pomoću alata Microsoft SQL Express Servera a nalazi se na Microsoft azure serveru.Da bi se mogli povezati s bazom podataka koristi se SqlConnection.

Za vezu se Azure SQL serverom potrebno je podesiti string za vezu s bazom.

String za vezu s bazom se kreira automatski te je podešen za vezu s bazom podataka koja se nalazi na lokalnom računalu.

Slika prikazuje ispravno podešen string za povezivanje s microsoft azure serverom.

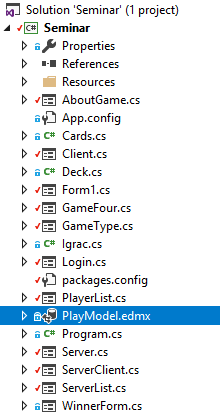
metadata=res://\*/PlayModel.csdl|res://\*/PlayModel.ssdl|res://\*/PlayModel.msl;provider=System.Data.SqlClient;provider connection string="data source=briscola.database.windows.net;initial catalog=Players;user id=želejno\_korisničko ime;password=željena\_šifra;MultipleActiveResultSets=True;App=EntityFramework"

Slika 3.1.4 String za konekciju s microsoft azure bazom podataka

3.2 Desktop aplikacija

U praktičnom dijelu rada kroz desktop aplikaciju demonstrirana je upotreba windows formi,entity okvira,json i tcp tehnologije. Aplikacija sadrži jedanaest formi te tri klase.

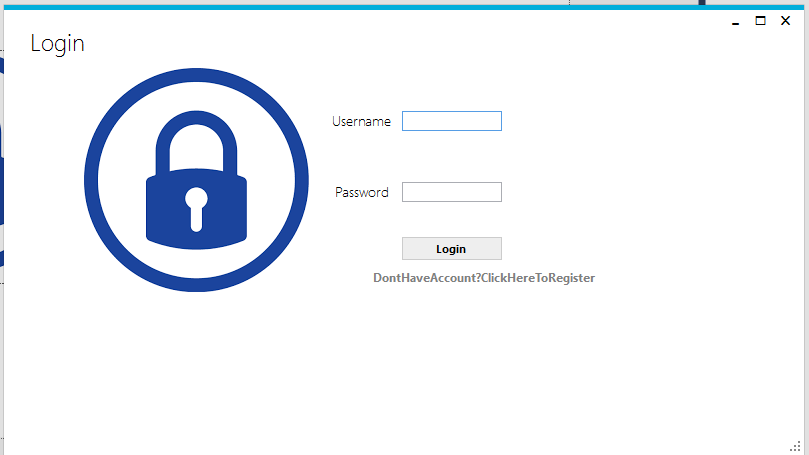
3.2.1 STRUKTURA APLIKACIJE



Slika 3.2.1 Struktura projekta

3.2.2 LOGIN FORMA

Pokretanjem aplikacije prikazuje se forma za prijavu( Login) koja omogućuje prijavu za postojeće korisnike,te registraciju novih korisnika.Klik na link registraciju odvodi korisnika na web aplikaciju Briscola.(Opisana u 3.3 poglavlju)

Nakon uspješnog logiranja,forma prelazi u pozadinu.Korisnik je usmjeren na glavni izbornik odnosno otvara se ServerClient forma,koji omogućava navigaciju korisnika.Satus korisnika iz nije spojen na mrežu (Offline) se mijenja u na mreži( Online) te se promjene pohranjuju u bazu podataka.

Slika 3.2.1 Login Forma

**3.2.2ServerClient forma**

****

**Slika 3.2.2. ServerClient forma sa admin pravima**

PrijestupnikKlijent (ServerClient) forma sadrži glavni izbornik.

Forma omogućuje navigaciju korisniku.

Izgled ServerClient forme je ovisan o pravima pristupa pojedinog korisnika.

Postoje dvije razine prava pristupa desktop aplikaciji a to su admin pristup i korisnički(User) pristup.

Kod Korisničkog pristupa ServerClient formi izbornik sadrži sljedeće opcije:

-pokretanje pristupnika(servera)

-podešavanje opcija pristupnika

-popis pokrenutih pristupnika

-upute za igru

ServerClient forma ima toolStrip koji sadrži mogućnosti za

-mystats-klikom miša prikazuje se statistika trenutno prijavljenoga korisnika.

-topPlayers-prikazuje se message box prozor,sadrži listu pet najboljih korisnika,sortiranih po postotku pobjeda te po broju odigranih igara.Ukoliku u bazi podataka ne postoji toliko korisnički računa,prikazat će se svi korisnici.

-homePage- klikom misa otvara se web aplikacija koja je dataljnije opisana u poglavlju 3.3

Admin pristup sadrži sve opcije koje sadrži korisnički(user) pristup prošireno za

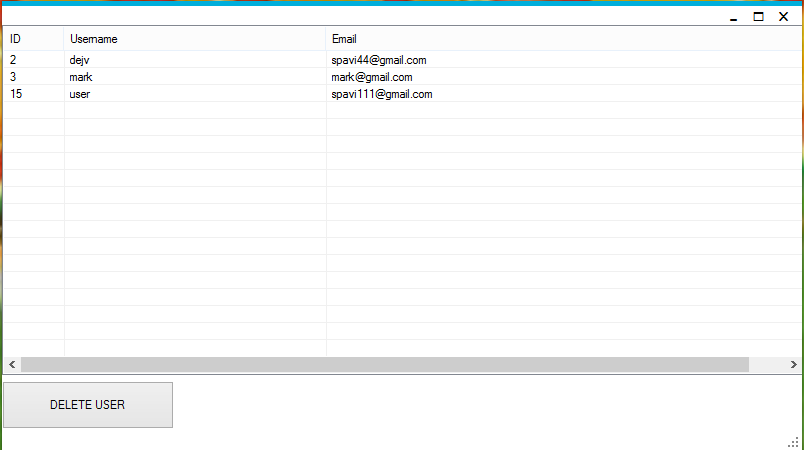
-listu igraća

-resetiranje baze

PlayerList forma

(Lista Igrača)PlayerList formi je moguće pristupiti samo s ulogom admina.

Forma sadrži popis svih korisnika s njihovim osnovnim informacijama.

Ova forma omogućuje adminu brisanje odabranih igrača iz baze podataka.

Slika 3 PlayerList forma

SERVERCLIENT

Klikom miša na neku od ponuđenih opcija otvara se odabrana forma, ServerClient Forma prelazi u pozadinu te njene kontrole više nisu vidljive korisniku sve dok se trenutna forma ne zatvori.ServerClient forma sadrži izbornik koji omogućuje navigaciju korisniku.

Izgled ServerClient je ovisan o pravima pristupa pojedinog korisnika.

Postoje dvije razine pristupa aplikaciji a to su administratorski pristup i korisnički pristup.

Prilikom korisničkog pristupa ServerClient izbornik sadrži sljedeće opcije: -pokretanje servera

-podešavanje opcija servera

-popis pokrenutih servera

-upute za igru

-toolStrip koji sadrži mogućnosti za

-mystats

-topPlayers

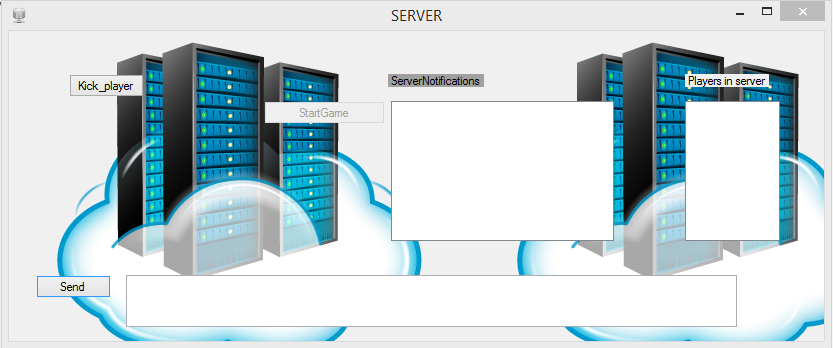
-homePage

Admin pristup sadrži sve opcije koje sadrži korisnik prošireno za

-listu igraća (Player List) opisano detaljnije

-resetiranje baze admin pristup dozvoljava resetiranje tablice igre(game) tj.uklanjaju se sve igre iz baze podataka,također gase se i svi trenutno pokrenuti serveri.

SERVER



Slika 7 Server

Pokretanjem server forme korisnik kreira tcp pristupnik(tcp server).Dohvaća se ip adresa korisnika ,zapisuje se u bazu podataka ,te se promjene pohranjuju.Prilikom pokretanja servera kreira se nova igra ,te se dodaje u bazu podataka .Ako se prilikom kreiranja pristupnik ne postavi vrsta igre(game type) opcije,kreirati će se igra s defaultnim vrijednostima:

-broj igrača:2 igrača

-broj karata:3 karte

Nakon starta tcp server sluša na portu 7000 zahtjeve klijenta za spajanje.Listview igraći na serveru(Players in server) sadrži popis korisnika koji se trenutno spojeni na pristupnik(tcp server).Lista klijenata spojenih na pristupnik se ažurira prilikom povezivanja klijenta na pristupnik ili odspajanja sa pristupnika.Pristupnik ima mogućnost da odabranoga klijenta odspoji sa servera klikom na botun Kick\_player,nakon ćega će klijent biti preusmjeren na Players list formu,te će biti uklonjen iz players in server listview-a.

Da bi se omogućila komunikacija s klijentom u pristupnik(Server) formi se treba obaviti pretplata na događaj DataReceived iz SimpleTcpServer klase.

tcpServer = new SimpleTcpServer();

tcpServer.Delimiter = 0x13;

tcpServer.StringEncoder = Encoding.ASCII;

tcpServer.DataReceived += Server\_Recived;

tcpServer.ClientConnected += Client\_Connected;

tcpServer.ClientDisconnected += Client\_Disconnect;

Slika 8.

U prikazanom kodu se vrši se pretplata na metode Client Connected,ClientDisconnected koje se nalaze u klasi SimpleTcpServe.Kako bi pristupnik(Server) bio obavješten o spajanju potrebno je obaviti pretplatu na Client Connected događaj.

Za obavjest o odspajanju klijenta sa pristpunika(Servera) potrebno je obaviti pretplatu na ClientDistonnected događaj.

Prilikom izlaska klijenta sa pristupnika(Servera),pristupnik će biti obaviješten o izlasku klijenta ,porukom u obliku message boxa.

Maksimalni broj klijenata koji mogu pristupiti serveru se postavlja u GameType formi, a po defaultnoj vrijednosti je postavljen na jednoga klijenta.

Nakon svakog spajanja/odspajanja klijenta na server , ažurira se broj klijenata na serveru te se promijene zapisuju u bazu podataka i pohranjuju.

Server nije u mogućnosti startati igru dok u client formi klijent ne klikne na ready botun.

Komunikacija između servera i klijena se odvija preko ugrađenoga chata koji se nalazi u server i client formi.

Prilikom pokretanja igre na server strani se izvršava serijalizacija objekta tipa objekt Form1 ili GameFour ovisno o vrsti igre.Serijalizirani objekt se u obliku stringa šalje klijentima kao poruka.

json = JsonConvert.SerializeObject(f); //serijalizacija Form1 objekta

tcpServer.BroadcastLine(json); //slanje json string svim klijentima spojenim na server

Slika 9

Nakon slanja json stringa klijentim a koji su spojeni na server izvršava se pretplata na sljedeće delegate

f.MouseClicked += this.OnMOuseClicked;

//mouse clicked se aktivira kada klijen klikne na pb1,2,3

//f.onmovemade rfferenca na metodu

f.GameEnded += this.OnGameEnded;

//on game ended obavještava klijenta o ishodu I salje unutar nje se poziva new game metoda

// game ended se aktivira kad prode 60 punti

this.MoveMade += f.OnMoveMade;

f.FormUnsub += this.OnFormUnsub;

f.FormExiting += this.OnFormClosing;

tcpServer.DataReceived += f.GameRecived;

Slika 10

Kako bi dobio obavijest o završetku igre server se pretplaćuje na event GameEnded klase Form1 ili GameFour ovisno o vrsti igre koja je pokrenuta.

Nakon što dođe do opaljivanja eventa Server obavještava klijenta o završetku igre, te poziva metodu NewGame(s).

Metoda NewGame(s) u Server formi se provjerava ishod igre te povećava broj pobjeda i poraza.

Dok ne dođe do konačnog ishoda igre prikazuje se dialog result gdje se nudi mogućnost nastavka igre.Ovaj dialog je nakon završetka prisutan i na server i na klijent strani.

DialogResult dialog = MessageBox.Show("Continue", "exit", MessageBoxButtons.YesNo);

Slika 11

U slučaju da je odgovor negativan (no) klijent će biti odspojen sa server.

Za poraženog proglasit će se onoga tko je prvi izabrao negativni odgovor.

OnFormUnsub("Form");

tcpServer.BroadcastLine("kicked");

Client\_Disconnect(null, tcpClient);

Slika 12

U slučaju da je odgovor “YES” započinje nova igra.

if (dialog == DialogResult.Yes)

{

//za izbjegavanje višestruke pretplate na događaj

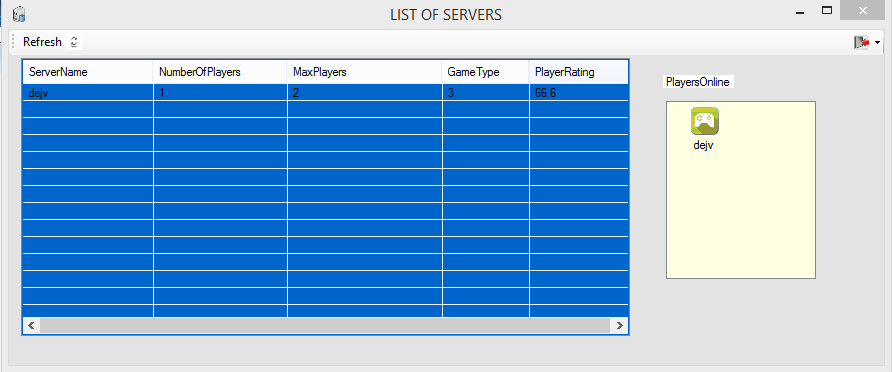
OnFormUnsub("Form");

}

Slika 13

Na slici 13 je prikazan kod gdje se kod pozitivnog odgovora poziva funkcija OnFormUnsub() u kojoj se vrši odjava sa pretplate sa svih događaja,

**Server list**

****

**Slika 14.**

ServerList forma sadrži popis podignutih servera koji su na raspolaganju korisniku.

Podaci se sadržani u ListView,te su prikazani detaljni podatci o serveru kao što su podaci o vrsti igre koja će biti kreirana na tom serveru.

ServerList forma sadrži popis igrača koji su trenutno prijavljeni u aplikaciji.

Korisnik odabire željeni server te dvoklikom miša pokušaje pristupiti serveru.

U slučaju uspješnoga povezivanja na server otvara se Client Forma,dok ServerList forma sakriti prelazi u pozadini i njene kontrole više nisu vidljive korisniku.

Do neuspješnoga povezivanja može doći u slučaju da je server popunjen,tada će se pojaviti MessageBox poruka s oddgovarajućom porukom.

///dodat mbox

Ako dođe do greške zbog problema s vezom ,tj. klijent se me može spojiti na server,program će baciti iznimku koja se će se prikazati u MessageBox obliku a sadržavat će odgovarajuću iznimku.

Client Forma



Slika 15. Client Forma

U slučaju uspješnoga povezivanja na server otvara se ClientForma.

Klijent ima mogućnost napuštanja servera pritisnkom na return botun,client forma se zatvara,te se ponovno pokreće ServerList forma.

ClientForma korisniku omogućuje komunikaciju sa serverom koristeći ugrađeni chat.

Da bi komunikacija sa serverom bila moguća klijent se mora pretplatiti na događaj iz SimpleTcpClient klase.

client = new SimpleTcpClient();

client.StringEncoder = Encoding.UTF8;

client.DataReceived += Client\_Received;

Slika 16.Client Incijalicazija

Kada je klijent spreman za pokretanje igre klikne spreman (ready) botun te čeka na odgovor servera.

U Client form se deserijalizira objekt tipa Form1 ili GameFour,koji je prethodno serijaliziran od strane servera te poslan kao string klijentu korištenjem tcp protokola.

BeginInvoke((MethodInvoker)delegate //delegate koji ne prima parametre te ima povratu vrijednost void

{

//da bi se uklonio delimiter “!!” koji tcp dodaje prilikom slanja

string s = start.Substring(0, start.Length - 1);

try

{

if (gameType == 3)

{

//deserijalizacija objekta tipa Form1

// DeserializeObject statička funkcija koja kao parameter prima string

f = JsonConvert.DeserializeObject<Form1>(s);

}

else

{

//deserijalizacija objekta tipa GameFour

g = JsonConvert.DeserializeObject<GameFour>(s);

}

Thread.Sleep(1);

}

catch (Exception e)

{

MessageBox.Show(e.Message.ToString());

}

}

Slika 17. Deseriajlizacija kod

Deserijalizaciju je potrebno izvršiti unutar BeginInvoke(MethodInvoker) funkcije,u suprotnom dolazi do greške tijekom izvođenja(runtime error).Form1 ili GameFour pokušat obaviti inicijalizacija elemenata prije nego se završi čitava deserijalizacija stringa.

Nakon deserijalizacije objekta potrebno je izvršiti pretplatu na događaje.

//Metoda OnMOuseClicked iz Client forme se pretplaćuje na događaj MouseClicked iz Forme1

f.MouseClicked += this.OnMOuseClicked;

this.ClientMove += f.OnClientMove;

client.DataReceived += f.GameRecived;

f.FormClosing += this.Game\_Form\_Closing;

Slika 18. Pretplata na događaje kod

Po završetku partije poziva se metoda OnFormUnsub(string s) . metodi se vrši odjava sa pretplačenih događaja.inače bi pri svakom pokretanju nove igre dolazilo da povećanja pretplate na event za jedan.

public void OnFormUnsub(string s)

{

if (gameType == 3)

{

this.MoveMade -= f.OnMoveMade;

f.GameEnded -= this.OnGameEnded;

f.MouseClicked -= this.OnMOuseClicked;

f.FormUnsub -= this.OnFormUnsub;

f.FormExiting -= this.OnFormClosing;

f.Close();

}

else

{

this.MoveMade -= g.OnMoveMade;

g.GameEnded -= this.OnGameEnded;

g.MouseClicked -= this.OnMOuseClicked;

g.FormUnsub -= this.OnFormUnsub;

g.FormExiting -= this.OnFormClosing;

g.Close();

}

}

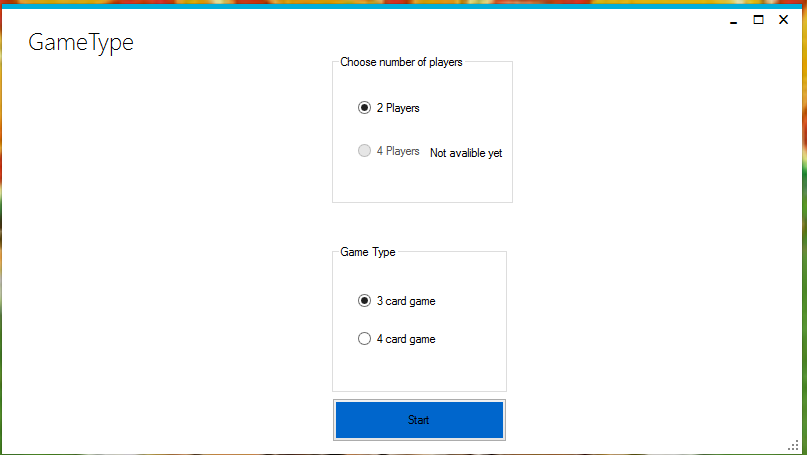
Slika 19.

GameType

GameType omogućuje korisniku koji kreira servera da namjesti željene postavke igre.

Broj igrača ,tj korisnika koji mogu pristupiti serveru,trenutno se ne može mijenjati,odnosno igra se može odvijati samo između dva korisnika.Game Type sadrži mogučnost odabira igre s tri ili s četiri karte.

U slučaju da se GameType forma ne pokrene koriste se defaultne postavke:2 igrača i 3 karte.



Slika 20

FORM 1

Form1 forma u kojoj se odvija sama igra briškola.



Slika 21 Form1

U From1 je implementirana više igrača „multiplayer“ logika.

**Multiplayer logika** se zasniva na tome je Form1 pretplaćena na događaj koji će se opaliti u server ili client formi ovisno o uloge igrača,tj dali je server ili klijent.

U Form1 sadrži MouseClicked event koji će se opaliti kada se klikne na neki od pitcureBoxova.Server i klijent su pretplaćeni na event MouseClicked .

U klijent formi će se izvršiti metoda ????? kada klijent klikne na PictureBox, PictureBox ili PictureBox.

U server formi će se izvršiti metoda

public void OnMOuseClicked(string s);

U server formi će se izvrsiti metoda

public void OnMOuseClicked(string s);

Kada server klikne na PictureBox1, PictureBox2 ili PictureBox3.

U server formi implementiran je MoveMade event koji će se opaliti kada klijent klikne na neki od PictureBox1,PictureBox2,PictureBox3.

Kako je Form1 pretplaćen na događaj MoveMade izvršit će se metoda

public void OnMoveMade(string s)

U client formi implementiran je ClientMove event koji će se opaliti kada server klikne na neki od PictureBox1,PictureBox2,PictureBox3.

Kako je Form1 pretplaćena na događaj ClientMove izvršit će se metoda:

public void OnClientMove(string s)

JsonObject(MemberSerialization.OptIn) izvan deklaracije klase nam omogućava da

korištenjem [JsonProperty] iznad svojstva odabiremo svojstva klase koja želimo serijalizirati.

U Form1 je implementirana metoda koji se koristi za računanje ukupnoga broja punti servera i klijenta.

U metoda **pokupi()** su implementirana pravila igre, te služi za povlačenje dijeljenje novih karata igračima.

**Pravila igre:**

Na početku igre svakome igraču su podijeljene po tri ili po četiri karte ovisno o vrsti igre.

Svaki igrač može odigrati kartu koju želi.Nije obavezno odgovarati na boju.Igrači jedan za drugim odigravaju karte a pobjednik ruke je onaj koji je bacio najjaču kartu.U slučaju da je odigrano više adutskih karti pobjednik je onaj igrač koji je odigrao najjaču adutsku karu.

Pobijedniku ruke se povečavaju bodovi ovisno o tome koliku vrijednost karata ruka sadrži.

Vrednovanje karata:11 bodova as,10 bodova trica,4 boda kralj,3 boda konj,2 boda fanat,7,6,5,4,2 ne donose bodove.

Pobjednik ruke prvi prediže novu kartu,te prvi igra slijedeću ruku.

Pobijedniku ruke se povečavaju bodovi ovisno o tome koliku vrijednost karata ruka sadrži.

Vrednovanje karata:11 bodova as,10 bodova trica,4 boda kralj,3 boda konj,2 boda fanat,7,6,5,4,2 ne donose bodove.

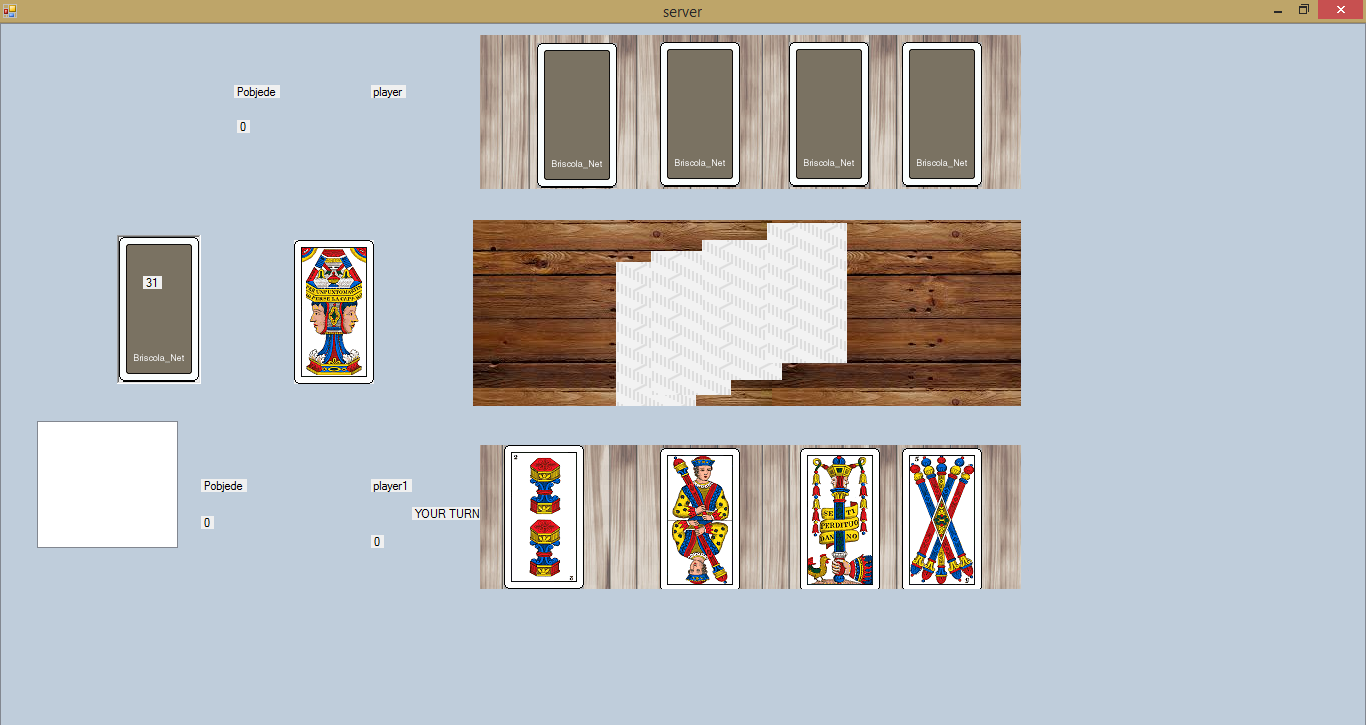
Pobjednik ruke prvi prediže novu kartu,te prvi igra slijedeću ruku.

Igra se odvija dok jedan od igrača nema više od 60 punata.U slučaju nerješenoga ishoda,ponovno se pokreće igra dok se ne dobije pobjednik.Kako bi se proglasio pobjednik potrebno je da jedan od igrača dođe do tri pobjede.U slučaju da jedan od igrača napusti igru dok je ista pokrenut,smatra se da je taj igrač poražen.

OnGameEnded(ishod\_igre)

GameFour

Da bi došlo do pokretanja ove forme potrebno je prilikom kreiranja server u game type formi promijeniti defaultnu opciju 3 cards,koja označava de će se prilikom pokretanja igre pokrenuti Form1.



Slika 22

Kroz ovu formu implementiran je tip igre gdje svaki igrač treba odigrati po dvije karte u svakoj ruci.

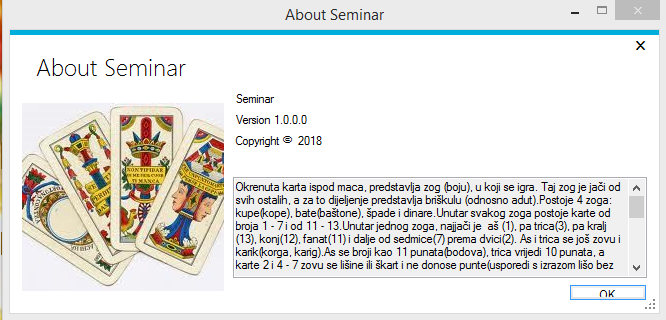
Multiplayer logika je ista kao kod Form1,uz preinake za dodatni picturebox4 i picturebox8.

Glavna razlika između GameFour i Form1 je u metodi **pokupi(),**jer da bi se odredio pobjednik ruke potrebno napraviti usporedbu između 4 karte za razliku od Form1 gdje su se uspoređivale samo dvije.

**AboutGame forma**

AboutGame Forma sadrži informacije o nazivu aplikacije,verziji,pravima,

autoru aplikacije,te pravila igre.



Slika 23

WinnerForm

Prikazuje se nakon završetka partije.Sadržaj forme ovisi o ishodu igre ,u slučaju pobjede korisnika prikazuje se win slika,dok u slučaju poraza prikazuje se lose slika.

Klasa Deck

Deck klasa korisiti se za izradu špila karata.

Svaka karta je opisana atributima vrijednost i vrsta.

Deck klasa koristi pobrojane vrijednosti Vrsta, Vrijednost.

public enum Vrsta { Kupe, Spade, Dinari, Bastoni };

public enum Vrijednost { dva, cetri, pet, sest, sedam, fanat, konj, kralj,tri,As };

Slika 24

Metoda setCards služi za izradu špila karata koji se korisiti u igri.

public void setCards()

{

int i = 0;

foreach (Vrsta v in Enum.GetValues(typeof(Vrsta)))

{

foreach (Vrijednost vr in Enum.GetValues(typeof(Vrijednost)))

{

string s = v.ToString() + vr.ToString(

this.karte[i] = new Cards(v, vr);

i++;

}

}

}

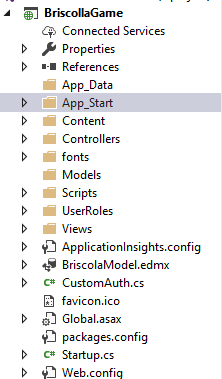
Deck klasa sadrži metodu **shuffle**() što jamči da će karte biti nasumično izmiješane prilikom svakoga pokretanja nove igre.

Klasa Igrač

Sastoji se od imena,punti,turn.

**Web Aplikacija**

Struktura



Slika 25

Iako ASP .NET MVC5 prvi radni okvir koji dolazi s Asp identity sustavom koji se brine za registraciju korisnika te prijavu postojećih. Za prijavu korisnika i registraciju korisnika u web aplikaciju izrađen je prilagođeni(custom) autentifikacijski sustav.

**Prava pristupa**

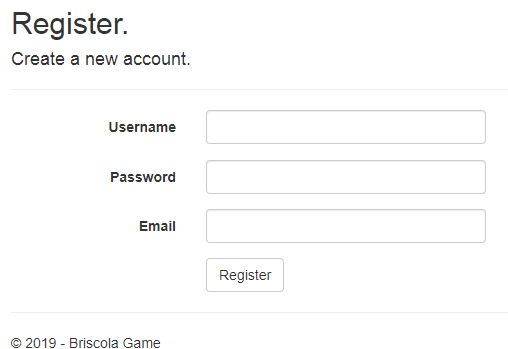
Prava pristupa su podijeljena na tri uloge:kao ne prijavljeni posjetitelj,kao korisnik,te kao admin.

Nelogirani korisnik se pri startu stranice preusmjerava login formu,te ima mogućnost pregleda home stranice,te about.

Korisnik ima mogučnost pregleda topplayers,mystats,playerlist,about,home,te može mijenjati svoje podatkke,tj email i username.

Administrator

**Registracija**



Slika 26

Nakon uspješne registracije korisnik može pristupi web i desktop aplikaciji.

Forma za registraciju sastoji se od polja za korisničko ime ,lozinku,email,te je izrađena na način da je jednostavno proširiti sa nekim od dodatnih polja kao sto su ponovni unos lozinke.

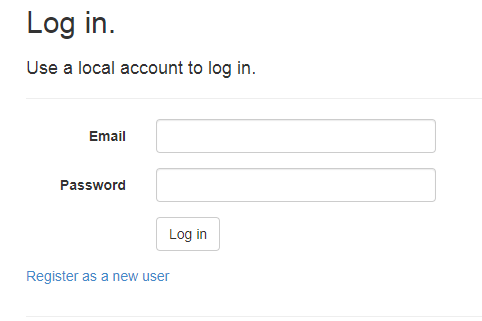
Nakon klika na registraciju provjerava se dali su zadovoljeni uvjeti,tj dali je lozinka minimalne veličine 6 znakova,te dali email adresa i korisničko ime već postoje u bazi podataka.

U slučaju neuspješne registracije,odnosno da nistu zadovoljeni gore navedeni uvjeti,posjetitelj će biti vraćen na register formu,te će se ispisati odgovarajuća poruka,koja kazuje koje polje nije zadovoljilo navedene kriterije.

Login

Forma za prijavu korisnika u web aplikaciju sastoji se od polja s email adresonm i lozinkom.

Prilikom startanja aplikacije, ukoliko korisnik nije prijavljen u aplikaciju,automatski se preusmjerava na login formu.Također korisnici koji ne posjeduju račun imaju mogućnost preko linka otvoriti formu za registraciju.

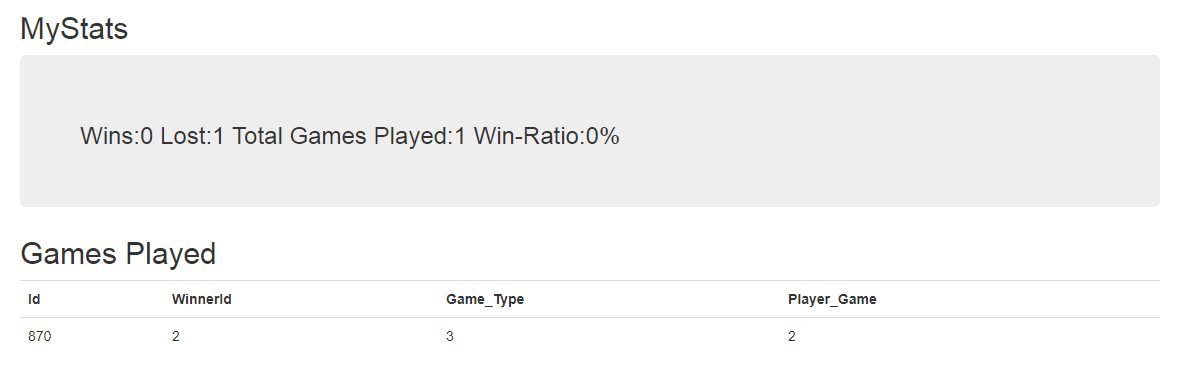


Slika 27

**My stats**

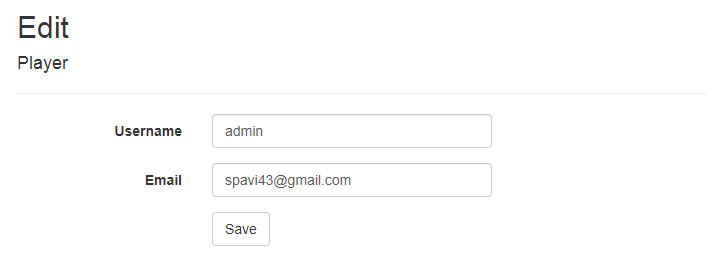
My stats prikazuje broj odigranih partija,broj pobjeda i poraza te ukupni postotak pobjeda,trenutno prijavljenog korisnika.

U games played dijelu sadržana je kompletna povijest igara u kojima je korisnik sudjelovao te ishod pojedine igre.

****

**Slika 27**

**Account Menager**



Slika 28

Logirani korisnici klikom na korisničko ime,koje se nalazi u navigaciji

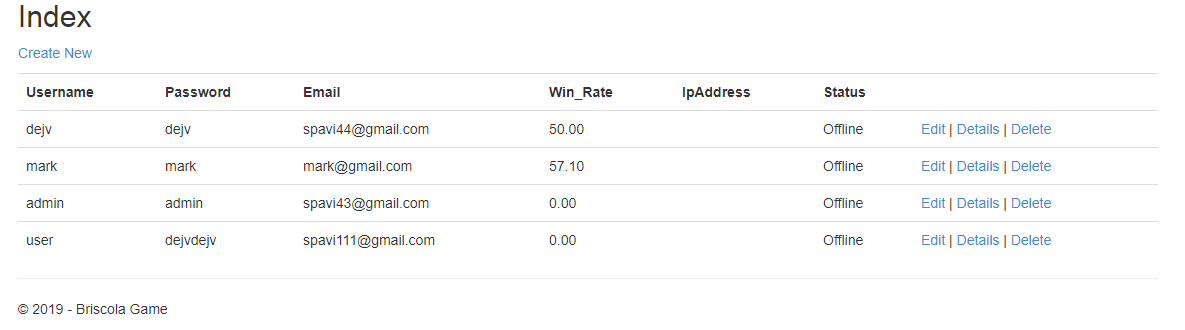
otvaraju account manager koji im omogućava promjenu korisničkog imena i email adrese.

Player List

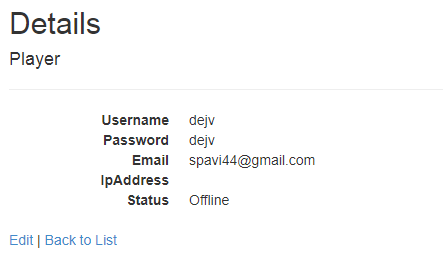
U player list(lista igrača) je sadržan popis svih korisnika, s povjerljivim informacijama kao što su lozinka i Ip adresa.Player List form ima postavljenu [Authorize(“Admin”)], što znači da

ovoj formi može pristupiti isključivo korisnik s pravom admin.

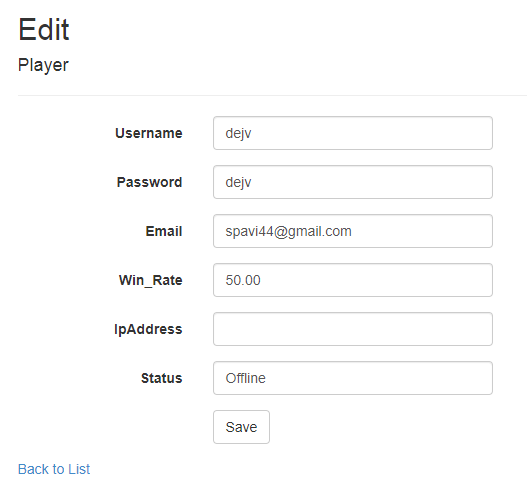
Prilikom pokušaja pristupa bez admin prava ispisuje se poruka o nedovoljnoj razini korisničkih prava.

Preko Player List forme admin može kreirati nove korisnike ,brisati postojeće,pregledati detaljne informacije o korisniku,te promijeniti neke od korisnikovih entiteta,kao npr. Status, ip adresa.

Slika 29



Slika 30 Forma Details za prikaz podataka korisnika

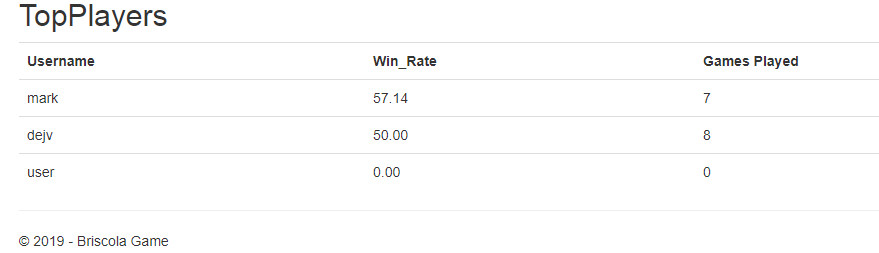


Slika 31

Prikaz forme za promjenu (Edit) podataka o korisniku.

**Top players**

Prikazuje listu najbolje rangiranih igrača prema povjesti odigranih igara.Lista sadrži najboljih pet igrača,ukoliko ima najmanje pet korisničkih računa u bazi podataka.U slučaju manjeg broja korisnika u bazi,lista će sadržavati sve korisnike koji se nalaze u bazi podataka.

Kriteriji rangiranja su po postotku pobjeda te po broju odigranih igara.Ovaj kriterij onemogućuje da igrač koji ima jednu pobjedu uz jednu odigranu igru, bolje rangiran od igrača koji ima manji postotak pobjeda ali mnogo veći broj odigranih igara.

Slika 32

public ServerList serverLista; //deklaracija varijable tipa ServerList

ZAKLJUČAK

Tema ovog završnoga rada bila je izrada multiplayer kartaške igre u desktop verziji sa pripadnom web aplikacijom.Ideja je bila izrada aplikacije koja se koristi u svrhu zabave korisnika .Deskotop aplikacija je izrađena korištenjem windows formi.

Web aplikacija je izrađena korištenjem aplikacijskog okvira ASP .NET MVC 5.

Desktop aplikacija i web aplikacija u pozadini koriste bazu podataka Igrači(Player) koja se nalazi na Microsoft azure serveru.

Glavni fokus je bio na izradi desktop aplikacije dok je web aplikacija dodana kako bi se administratoru omogućilo jednostavnije upravljanje bazom podataka,dok je korisnkiku omogućena izmjenu osobnih podataka,te pregled povijesti igara.Web aplikacija omogućava jednostavniji pristup bazi time što nije potrebno imati desktop aplikaciju.Posjetitelju je u time omogućen i mobilni pristup bilo gdje u svjetu.

C# kombiniran s windows formama koje su dio.NET tehnologije predstavlja razvojno okruženje koje omogućava razvoj bilo kojega tipa aplikacija,ćime se pokrivaju svi zahtjevi tržišta.

Izbor programskog jezika sveo se na C# zbog njegove jednostavnosti te velike lepeze mogućnosti.

Razvoj aplikacije se nastavlja.U budućnosti će desktop aplikacija će dobiti nove funkcionalnosti poput mogućnost prisustvovanja do četiri korisnika u igri,te će korisnik moći birati dali će igrati protiv korisnika ili protiv računala.Web aplikacija će dobiti funkcijonalnosti koje će omogućiti pokretanje igre iz web preglednika.

5.Literatura

<https://www.c-sharpcorner.com/article/from-zero-to-hero-in-json-with-c-shar/>

<https://www.softwaretestinghelp.com/create-json-objects-using-c/>

<https://www.geeksforgeeks.org/socket-programming-in-c-sharp/>

<https://www.c-sharpcorner.com/article/socket-programming-in-C-Sharp/>

<https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2008/dd30h2yb(v%3dvs.90)>

<https://code.msdn.microsoft.com/windowsdesktop/Communication-through-91a2582b>

<https://adrianscorner.wordpress.com/2014/04/04/designing-a-many-to-many-relationship-with-additional-fields-using-entity-framework/>

<https://www.tutorialsteacher.com/mvc/asp.net-mvc-tutorials>

<https://magliveli.files.wordpress.com/2010/11/1-8-programming-c.pdf>

<https://www.entityframeworktutorial.net/>

<https://docs.microsoft.com/en-us/ef/ef6/modeling/code-first/workflows/existing-database>

<https://urn.nsk.hr/urn:nbn:hr:200:800058>

<http://mreze.layer-x.com/s010200-0.html>

6.POPIS SLIKA

Slika 1 Struktura desktop aplikacije...............................................................................

Slika 2 Server Client Forma admin prava.......................................................................

Slika 3 Player List forma ......................................................................................

Slika 4 Kod konekcija sa bazom ........................................................................................

Slika 5 EF baza prava schema........................................................................................

Slika 6 Baza podataka izgled .........................................................................................

Slika 7 Login forma .........................................................................................

Slika 8 Server forma .........................................................................................

Slika 9 .........................................................................................

Slika 10 .........................................................................................

Slika 11 .........................................................................................

Slika 12 .........................................................................................

Slika 13 .........................................................................................

Slika 14 .........................................................................................

Slika 15 .........................................................................................

Slika 16 .........................................................................................

Slika 17 ........................................................................................................................................

Slika 18 ........................................................................................................................................

Slika 19 ........................................................................................................................................

Slika 20 ........................................................................................................................................

Slika 21 ........................................................................................................................................

Slika 22 ........................................................................................................................................

Slika 23 ........................................................................................................................................

Slika 24 ........................................................................................................................................

Slika 25 ........................................................................................................................................

Slika 26 ........................................................................................................................................

Slika 27 ........................................................................................................................................

Slika 28 ........................................................................................................................................

Slika 29 ........................................................................................................................................

Slika 30 ........................................................................................................................................

Slika 31 ........................................................................................................................................

Slika 32 ........................................................................................................................................

Slika 33 ........................................................................................................................................

Slika 34 ........................................................................................................................................