DESKTOP APLIKACIJA ZA PRAĆENJE REZULTATA KOŠARKAŠKE UTAKMICE

Sokol, Matija

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:282083

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-11-30**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

DESKTOP APLIKACIJA ZA PRAĆENJE REZULTATA KOŠARKAŠKE UTAKMICE

Završni rad

Matija Sokol

Osijek, 2019.

SADRŽAJ

1. UVOD 1
    1.1. Zadatak završnog rada 1
2. KORIŠTENE TEHNOLOGIJE 2
    2.1. Programski jezik C# 2
    2.2. Microsoft Visual Studio 3
3. POSTOJEĆA RJEŠENJA 5
    3.1. Easy Stats for Baskteball 5
    3.2. TurboStats 6
4. PROGRAMSKO RJEŠENJE 7
    4.1. Dijagram tijeka 7
    4.2. Struktura projekta 8
    4.3. Forma za animaciju 10
    4.4. Početna forma 10
    4.5. Forma za dodavanje i brisanje ekipa 12
    4.6. Forma za dodavanje igrača nove ekipe 14
    4.7. Forma za pregled rezultata 16
    4.8. Forma za postavljanje detalja utakmice 17
    4.9. Forma za tijek utakmice 18
    4.10. Nedostatci aplikacije i mogućnosti za nadogradnju 21
5. ZAKLJUČAK 22
LITERATURA 23
SAŽETAK 25
ABSTRACT 26
ŽIVOTOPIS 27

1. UVOD

U današnje vrijeme, kada većina populacije u svijetu svakodnevno koristi računala zbog olakšanog pristupa rješavanja zadataka, potrebno je ponuditi što više programa koji će omogućiti isto. Jedan od takvih programa je i aplikacija koja je izrađena kao tema ovog završnog rada.

Cilj ovog rada bio je izraditi računalnu aplikaciju koja će omogućiti jednostavno praćenje rezultata košarkaške utakmice. Pri tome se misli na odabir ekipa koji igraju, postavljanju karakteristika utakmice (trajanje četvrtine, broj prekida (engl. *time-out*)) te na kraju i sam tijek utakmice koji korisnik može kontrolirati. Također, omogućeno je dodavanje novih ekipa tako da je aplikacija prilagođena i za korištenje u vlastite svrhe.

U izradi aplikacije korišten je programski jezik C# te razvojno okruženje (engl. *development environment*) Microsoft Visual Studio. Kako bi se aplikacija uspješno izradila, bilo je potrebno dodatno proširiti znanje iz već navedenog programskog jezika.

1.1. Zadatak završnog rada

U okviru ovog završnog rada potrebno je upoznati se sa C# programskim jezikom, Windows formama te radu s više formi. Potrebno je izraditi aplikaciju s grafičkim sučeljem koristeći gore spomenute tehnologije koja će omogućiti korisniku praćenje tijeka košarkaške utakmice što uključuje rezultat, vrijeme, bilježenje igrača koji je postigao koš, prekršaje te stanke. Korisnik iz izbornika odabire i učitava ekipe koje igraju (popis igrača, logo ekipe). Aplikacija mora omogućiti praćenje rezultata uz mogućnost pohrane rezultata u datoteku kao i učitavanje prethodnih rezultata iz datoteke.

2. KORIŠTENE TEHNOLOGIJE

Kao što je napisano u uvodu, za izradu završnog rada korišten je programski jezik C# te razvojno okruženje Microsoft Visual Studio 2019 Community. Microsoft omogućuje besplatno korištenje ovog izdanja studentima. Visual Studio je dobro poznat svim studentima jer se tijekom fakultetskog obrazovanja puno kolegija koriste upravo njime. Što se tiče samog programskog jezika, tijekom nastave dane su samo osnove i uvod u isti pa je bilo potrebno samostalno proučiti dodatnu literaturu i upoznati se s nekim od naprednih funkcionalnosti.

2.1. Programski jezik C#

C# je objektno orijentirani programski jezik [1] kojeg su razvili Anders Hejlsberg i drugi zaposlenici tvrtke Microsoft. Strogo je tipiziran jezik koji se oslanja na Microsoftov .NET radni okvir (engl. *framework*) kako bi omogućio izradu različitih programskih rješenja, od klasičnih računalnih (engl. *desktop*) aplikacija, preko web servisa do aplikacija za oblak (engl. *cloud*) platforme. Budući da je u potpunosti objektno orijentiran, svaka varijabla ili funkcija mora biti smještena u klasu. Za razliku od C++ programskog jezika, koristi skupljanje smeća [2] (engl. *garbage collection*). Time se omogućuje automatsko oslobađanje zauzetih resursa jednom kada oni nisu potrebni. Programski kod ne prevodi se izravno u strojni kod, već se najprije prevodi u IL (engl. *intermediate language*) i učitava u CLR (engl. *common language runtime*), nakon čega ga JIT (engl. *just in time*) prevoditelj prevodi u strojni kod. Početni kod nakon stvaranja novog projekta prikazan je izlistanjem koda 2.1.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

Izlistanje koda 2.1. Početni kod nakon stvaranja novog projekta

.NET platforma je razvojni okvir koji omogućuje novo sučelje za programiranje aplikacija (engl. *application programming interface*, API) i ujedinjuje klasično sučelje Windows

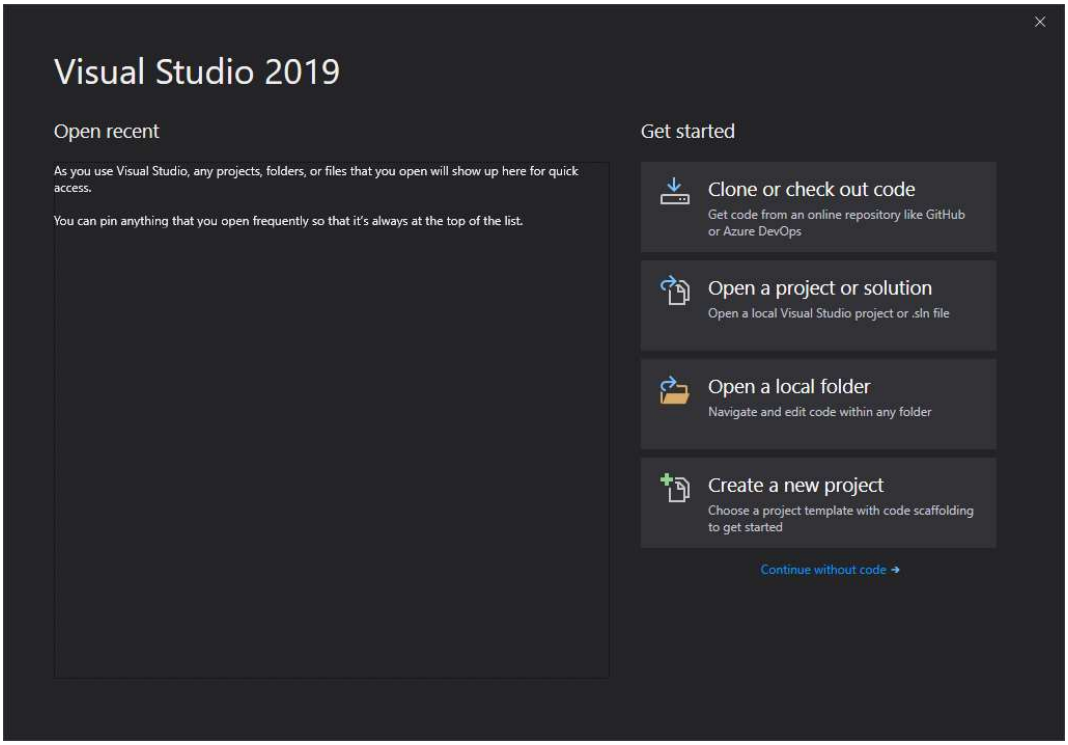
operacijskog sustava, zajedno s brojnim tehnologijama koje su proizašle iz Microsoft-a, kao što su ASP (engl. *active server pages*) razvojni okvir za web, XML (engl. *extensible markup language*), objektno orijentirani dizajn i drugo.

C# sadrži sve dobre odlike potpuno objektno orijentiranog jezika (koje većinom preuzima iz C++ i Jave), a u sklopu .NET platforme omogućava kreiranje vizualnih aplikacija čak i onim korisnicima koji nemaju programerskog iskustva. Također, daje dobar uvid u način na koji nastaju objektne i vizualne aplikacije i vrlo je lako naučiti korisnike kako projektirati takve aplikacije i upravljati njihovim korištenjem.

2.2. Microsoft Visual Studio

Microsoft Visual Studio [3] je integrirano razvojno okruženje (engl. *integrated development environment*, IDE) kojeg je razvila tvrtka Microsoft. Koristi se za razvoj računalnih aplikacija, internet stranica i servisa te mobilnih aplikacija.

Prva verzija Visual Studia izdana je 1997. godine pod nazivom Visual Studio 97. Prije toga, Visual Basic, Visual C++, Visual FoxPro i Visual SourceSafe su se prodavali kao zasebni proizvodi. Danas je najnovija verzija Visual Studio 2019. Početni zaslon Microsoft Visual Studia 2019 prikazan je slikom 2.1.

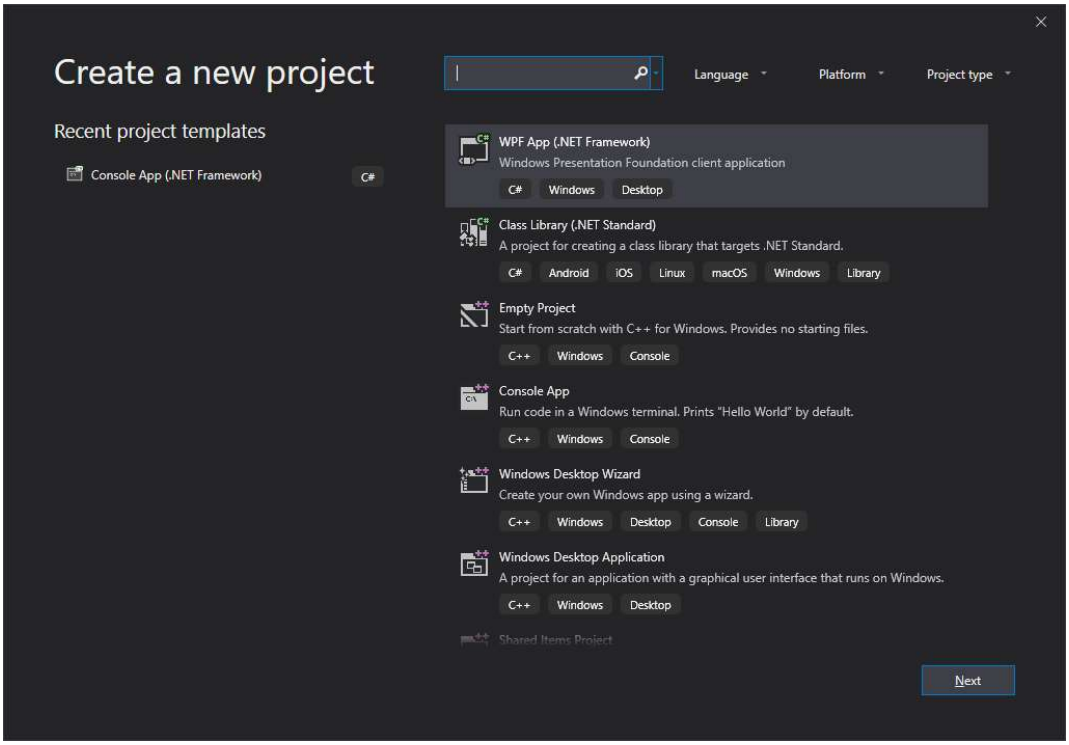


Slika 2.1. Početni zaslon Microsoft Visual Studia 2019

Postoje tri izdanja ovog proizvoda. Ovisno o potrebi, korisnik može upotrebljavati Community, Professional ili Enterprise izdanje. Za razliku od Community-a koji je besplatan za akademske potrebe, Professional pruža integrirano razvojno okruženje za sve podržane razvojne jezike. Enterprise, osim što pruža sve stavke Professional izdanja, nudi mogućnosti novih dijelova za razvijanje baze podataka, surađivanja te alata za testiranje i prijavljivanje grešaka. Community izdanje dostupno je, kao i još nekolicina proizvoda, preko Microsoftovog sustava DreamSpark.

Visual Studio sadrži uređivač izvornog koda koji podržava IntelliSense (komponenta koja predlaže ostatak teksta) kao i refaktoriranje koda. Također, sadrži alate koji se koriste za izradu aplikacija s grafičkim korisničkim sučeljem (engl, *graphical user interface*, GUI) te za uređivanje internet stranica.

Podržava 36 različitih programskih jezika te dozvoljava uređivaču koda i alatu za uklanjanje grešaka (engl. *debugger*) da podržava gotovo bilo koji programski jezik, pod uvjetom da usluga za taj jezik postoji. Na slici 2.2. prikazan je zaslon stvaranja novog projekta.



Slika 2.2. Stvaranje novog projekta

3. POSTOJEĆA RJEŠENJA

Pretražujući internet pronađene su brojne aplikacije koje pružaju sličnu funkcionalnost kao i aplikacija razvijena u ovom radu. Budući da je u današnje vrijeme vrlo popularna izrada mobilnih aplikacija, većina pronađenih aplikacija je upravo namijenjena za tu vrstu platforme, dok je manji broj namijenjen za računala. Neke od njih će bit opisane u nadolazećim potpoglavljima.

3.1. Easy Stats for Basketball

Aplikacija [4] Easy Stats for Baskteball namijenjena je za iOS operacijski sustav. Omogućuje jednostavno praćenje rezultata te je moguće podijeliti sadržaj na nekoj od društvenih mreža, na primjer Facebook, Twitter ili jednostavno proslijediti poveznicu.

Vrlo je korisna za trenere koji moraju detaljnije pratiti učinke svojih igrača. Statistiku protivničkih igrača nije potrebno unositi jer se oni promatraju kao cjelina, odnosno tim.

Neke od karakteristika koje je potrebno naglasiti su da je omogućeno beskonačan broj utakmica, ekipa i igrača u besplatnoj inačici. Također postoji prikaz tablice u kojoj je navedena statistika, slično kao i na službenim internetskim stranicama za praćenje košarkaških utakmica. Korisnik može vidjeti tijek utakmice pomoću dijagrama toka te sortirati popis igrača prema različitim kriterijima. Izgled aplikacije vidljiv je na slici 3.1.

Slika 3.1. Izgled aplikacije Easy Stats for Basketball

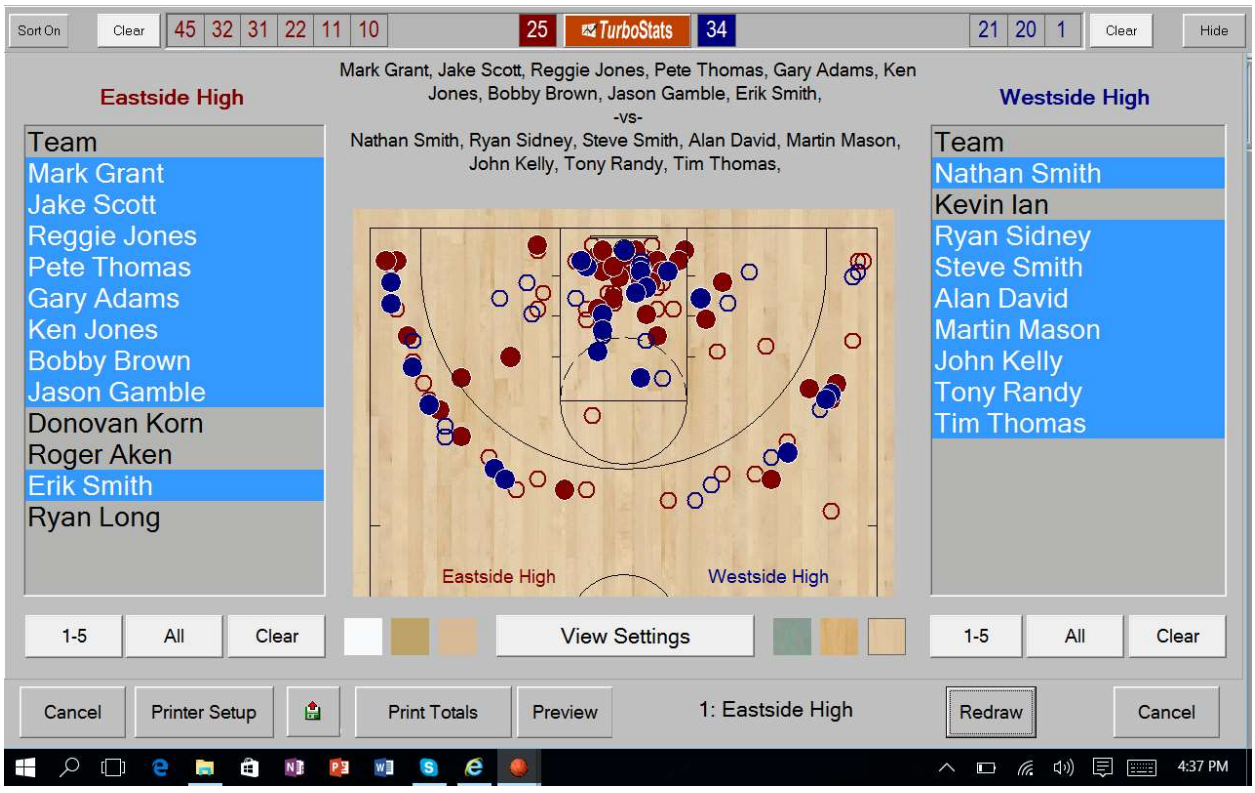
3.2. Turbo Stats

Aplikacija [5] Turbo Stats namijenjena je korištenju na Windows platformi. Nije namijenjena besplatnom korištenju, ali postoji besplatni probni rok u kojemu svatko može isprobati kako aplikacija radi.

Omogućeno je praćenje više od sto vrsta statistika, bodova po posjedu, grafove te statistiku po minuti. U aplikaciji postoji prikaz košarkaškog igrališta na kojemu se prikazuju podaci o ubačajima nakon izgubljene lopte, drugim prilikama u napadu, kontrama.

Aplikacija pruža mogućnost simuliranja napada protiv različitih vrsta obrana. To uključuje prikaz igrača na pozicijama koji se kreću ovisno kako akcija treba funkcionirati.

Svi podaci se mogu spremati na oblak kako bi se isti mogli dijeliti zbog lakše komunikacije među trenerskim osobljem. Na slici 3.2. prikazan je izgled aplikacije.



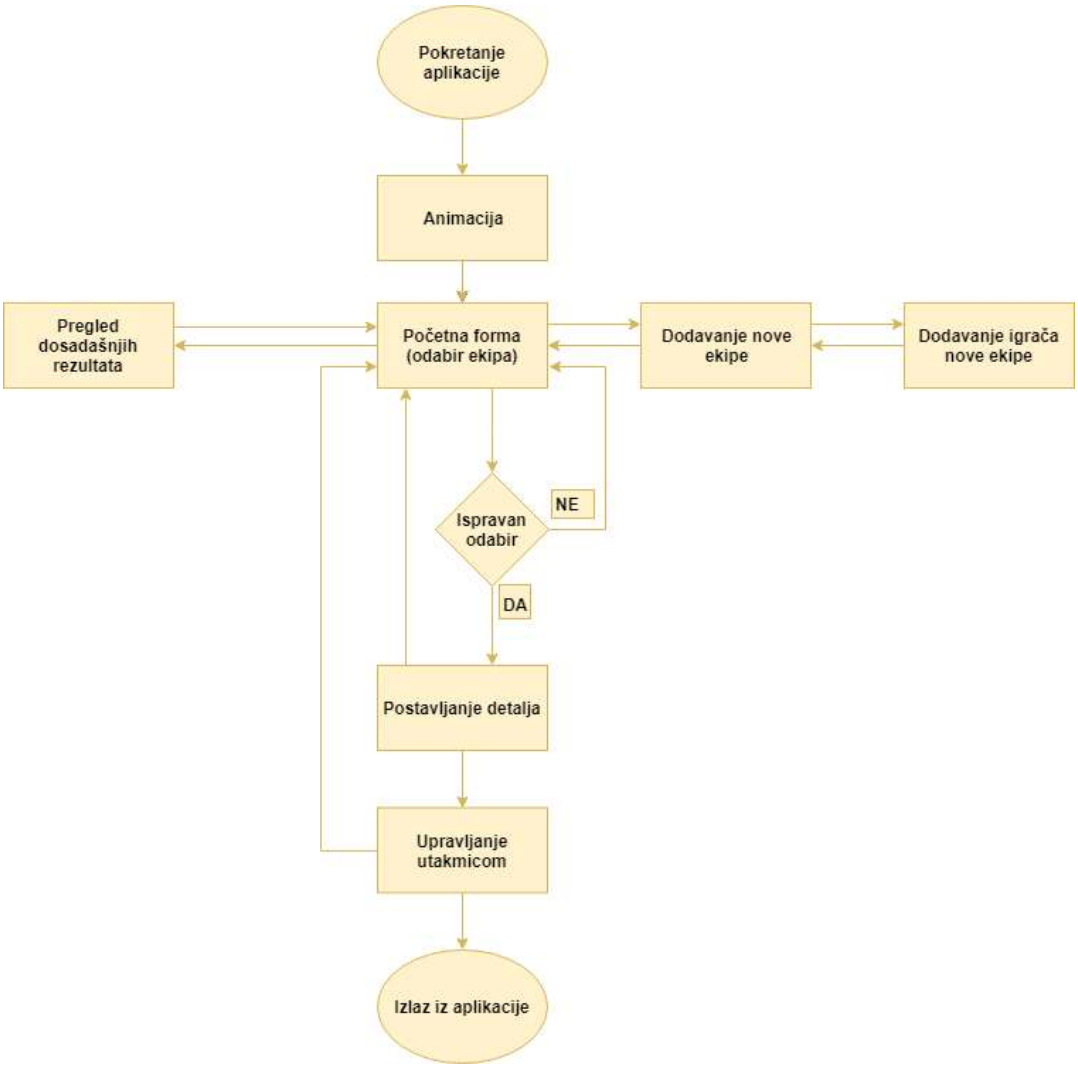
Slika 3.2. Izgled aplikacije Turbo Stats

4. PROGRAMSKO RJEŠENJE

U daljnjem tekstu će biti opisano što se i kako odvija u implementiranom programskom rješenju. Cijela aplikacija se sastoji od 7 formi, od kojih je svaka zadužena za određenu funkcionalnost. To su forme za početnu animaciju, odabir ekipa, dodavanje i brisanje ekipe, dodavanje igrača nove ekipe, pregled dosadašnjih rezultata, postavljanje detalja utakmice i tijek utakmice.

4.1.Dijagram tijeka

Budući da je svaki vizualni prikaz, u ovom slučaju dijagram tijeka, ugodan i učinkovit način prezentiranja programskog rješenja, na slici 4.1. može se vidjeti na koji način funkcionira aplikacija.



Slika 4.1. Dijagram tijeka aplikacije

Iz dane slike vidljivo je kako iz određene forme doći do neke druge. Kod koji služi za navedeno je poprilično jednostavan. Potrebno je samo stvoriti novi objekt određene forme te pozvati metodu *Show()* ili *ShowDialog()* na istom. Ovisno o potrebi, biramo između te dvije metode. Prva omogućuje upravljanje prethodnom formom, ukoliko ona nije zatvorena, dok druga to ne dopušta. Korištenje ovih metoda prikazano je izlistanjem koda 4.1. i 4.2.

```
MainForm mainForm = new MainForm();
mainForm.Show();
```

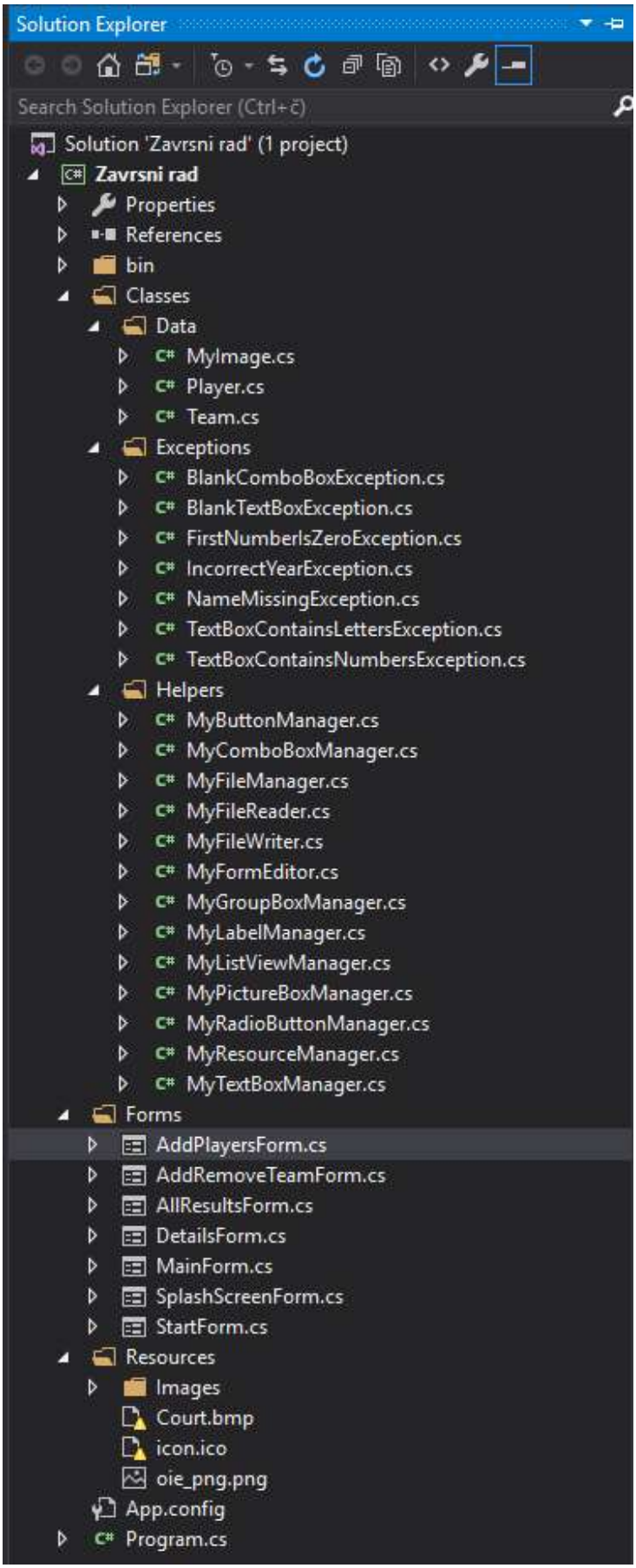
Izlistanje koda 4.1. Otvaranje nove forme metodom *Show()*

```
AddTeamForm addTeamForm = new AddTeamForm();
addTeamForm.ShowDialog();
```

Izlistanje koda 4.2. Otvaranje nove forme metodom *ShowDialog()*

4.2. Struktura projekta

Svaki veći i ozbiljniji projekt se nakon nekog vremena poprilično „zatrpa“ s datotekama klasa i ostalim dijelovima. Na manjim projektima to nije vidljivo te se u većini slučajeva kod ne organizira u pakete, to jest direktorije. Da bi se programeru olakšao rad u razvojnom okruženju, Visual Studio omogućuje stvaranje novih paketa te se srodne datoteke mogu grupirati kako bi kod bio pregledniji i kako bi se datoteke lakše pronašle. Struktura ovog projekta vidljiva je na slici 4.2.



Slika 4.2. Struktura projekta

Na slici se može primijetiti da ima puno datoteka u *Helpers* direktoriju. One su napravljene kako bi se kod iz klasa forme izdvojio van njih te tako smanjio veličinu istih. Svaka kontrola ima svoju klasu koja omogućuje funkcionalnosti samo za nju.

4.3. Forma za animaciju

Forma je osmišljena s ciljem da se pojača vizualni dojam aplikacije. Možda se naizgled programski kod koji se u pozadini odvija čini kompliciran, on je, upravo suprotno, poprilično jednostavan.

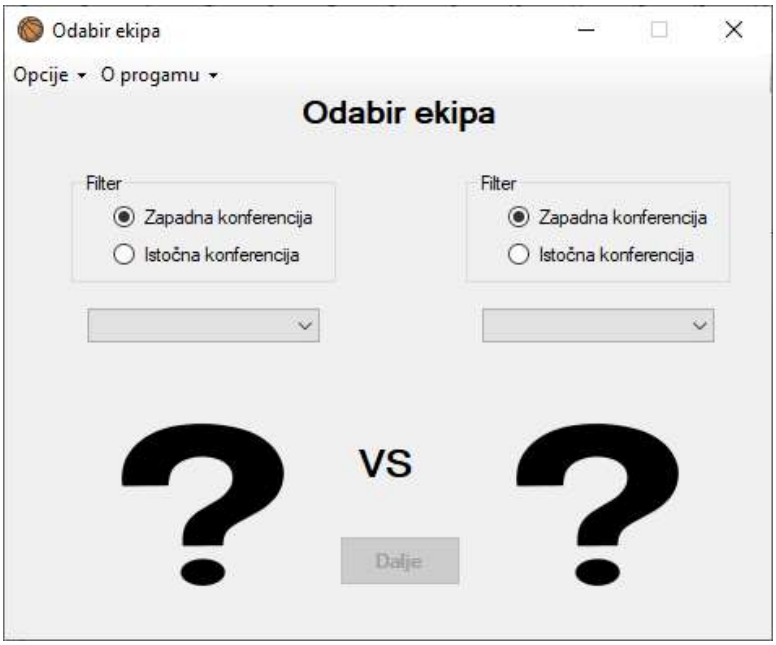
Slika 4.3. Izgled forme za animaciju

Na formu su, kao što je prikazano slikom 4.3., dodana dva elementa za slike, te su dinamički postavljeni da prikazuju slike [6][7] gif (engl. *graphics interchange format*) formata. Kako bi se forma ugasila nakon određenog vremena, bilo je potrebno koristiti element za praćenje vremena (engl. *timer*). Na istome je postavljen interval, odnosno, vrijeme koliko dugo forma za animaciju traje. Potrebno je naglasiti da je mjerna jedinica za interval milisekunda.

4.4. Početna forma

Glavna svrha početne forme je da se odaberu ekipe koje će odigrati utakmicu. Kako bi se to omogućilo, na formi su korišteni neki od osnovnih elemenata koji predstavljaju tipke (engl. *Button*), slike (engl. *PictureBox*), radijske gumbe (engl. *RadioButton*), tekstualne zapise (engl. *Label*), izbornike s padajućim izborom (engl. *ComboBox*) i tako dalje. Budući da svi ti elementi

nasljeđuju osnovnu klasu *Control*, možemo ih zvati kontrolama te im isto tako i pristupiti [8]. Naslovna traka svake forme sadrži logo aplikacije [9] te naziv, što je vidljivo na slici 4.4.



Slika 4.4. Izgled početne forme

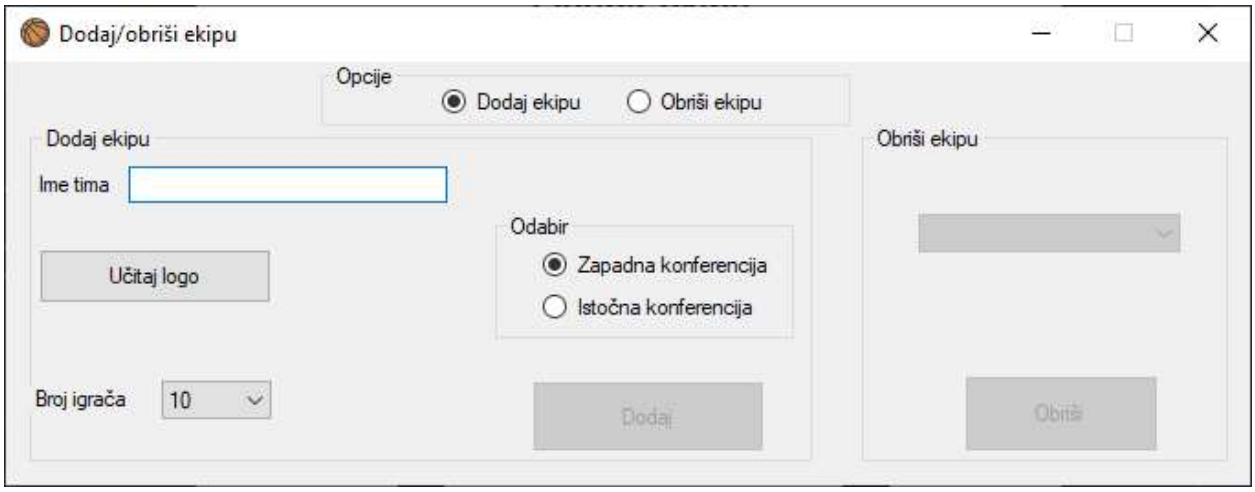
Podaci za ekipe se učitavaju iz datoteke te se spremaju u listu. Nakon toga, popunjava se element s padajućim izbornikom. Na temelju odabira iz navedenog elementa, prikazuje se slika koja predstavlja logo ekipe. Kao što je i vidljivo sa slike, gumb koji otvara iduću formu je onemogućen sve dok podaci nisu ispravni. U ovom slučaju to znači da iste ekipe ne mogu igrati jedna protiv druge. Izlistanjem koda 4.3. prikazano je čitanje podataka iz datoteke na način da se provjerava postoji li linija te se ista sprema u listu.

```
public static List<string> ReadFromFile(string path)
{
    List<string> list = new List<string>();

    if (File.Exists(path))
    {
        using (StreamReader reader = new StreamReader(path))
        {
            string line;
            while ((line = reader.ReadLine()) != null)
            {
                list.Add(line);
            }
            reader.Close();
        }
    }

    return list;
}
```

Izlistanje koda 4.3. Metoda za čitanje podataka iz datoteke

4.5. Forma za dodavanje i brisanje ekipa

Kao što je u uvodu opisano, aplikacija je namijenjena za vlastite potrebe. Pod tim se podrazumijeva da svatko tko koristi aplikaciju može napraviti, odnosno dodati vlastitu ekipu te igrače. Na samom vrhu vidljiva su dva radijska gumba koja omogućuju, to jest onemogućuju kontrole za dodavanje, odnosno brisanje ekipa, kao što je prikazano na slici 4.5.



Slika 4.5. Izgled forme za dodavanje i brisanje ekipa

Za ovaj dio aplikacije bilo je potrebno upoznati se s klasom *File* [10] koja omogućuje puno metoda za upravljanje datotekama na disku računala. Nakon što se forma otvori, korisnik je u mogućnosti unijeti ime i logo ekipe [11]. Budući da nije dozvoljeno imati više ekipa s istim imenom, tipka za dodavanje je onemogućena sve dok unos nije ispravan. Klikom na tipku „Učitaj logo“ otvara se novi prozor u kojemu korisnik odabire logo ekipe. Omogućeno je filtriranje različitih vrsta slikovnih datoteka radi lakšeg pronalaženja tražene slike. Ovakav način učitavanja slike prikazan je izlistanjem koda 4.4.

```
private void ButtonLoadLogo_Click(object sender, EventArgs e)
{
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Filter = "Slikovne datoteke|*.png;*.jpg|PNG|*.png|JPEG|*.jpg";
    ofd.Title = "Odaberi sliku ekipe";

    if (ofd.ShowDialog() == DialogResult.OK)
    {
        FILE_PATH = ofd.FileName;
        FILE_NAME = ofd.SafeFileName;
        imageUploaded = true;
        TextBoxTeamName_TextChanged(sender, e);
        pictureBoxPreview.Image = Image.FromFile(FILE_PATH);
    }
}
```

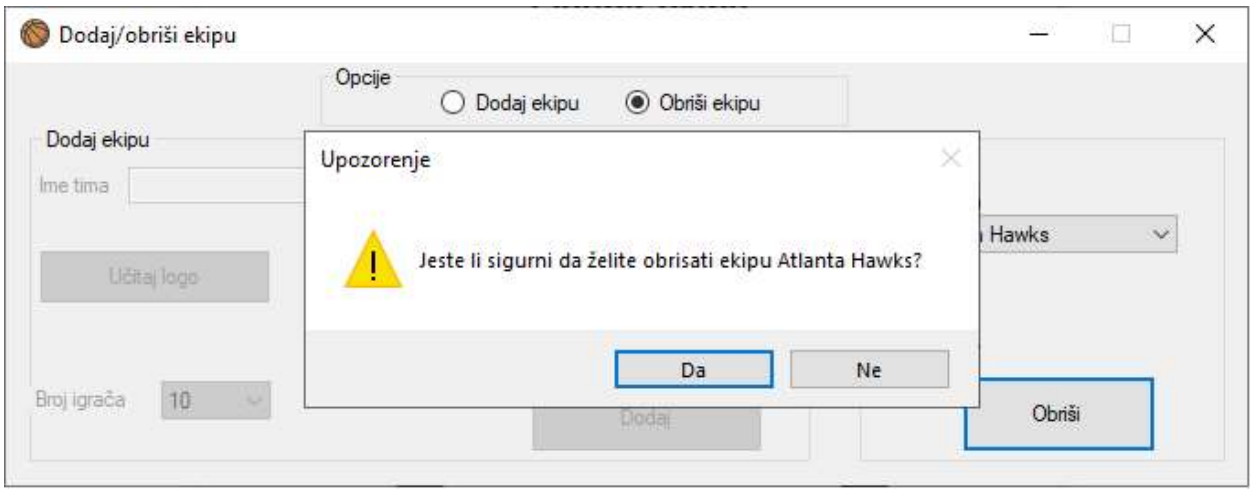
Izlistanje koda 4.4. Metoda za klik na „Učitaj logo“ tipku

Također, svaki tim mora biti svrstan u svoju konferenciju, pa je omogućen i odabir za to. Kao što je vidljivo sa slike, ekipa ne mora nužno imati deset igrača, već je ponuđeno osam ili devet. Forma za dodavanje igrača će biti formirana baš na temelju tog podatka, no o tome će biti više riječi u idućem potpoglavlju. Svaka učitana slika sprema su poseban direktorij gdje se nalaze slike svih ekipa. Izlistanjem koda 4.5. prikazan je način kopiranja slike koja se učitava već spomenutom metodom.

```
public void CopyImage()
{
    string destinationPathForImage = Path.Combine(Application.StartupPath, "Images",
"Teams Logos", FILE_NAME);
    File.Copy(FILE_PATH, destinationPathForImage);
    MyFileManager.RenameFile(textBoxTeamName, FILE_PATH, FILE_NAME, FILE_NAME);
}
```

Izlistanje koda 4.5. Metoda za kopiranje slike u direktorij sa ostalim slikama

Budući da je implementirano dodavanje novih ekipa, bilo je potrebno i omogućiti brisanje postojećih. Ova funkcionalnost je bila puno lakša za napraviti jer, da bi se ekipa obrisala, trebalo je samo obrisati logo ekipe, popis igrača te ime ekipe s popisa ekipa. Za navedeno je potrebno znati jedino putanju datoteka na disku, no kako su iste spremljene u direktorij projekta, nije bilo problema doći do njih.



Slika 4.6. Izgled forme nakon klika na tipku „Obriši“

Kako se ne bi dopustilo slučajno brisanje, to jest slučajni klik na tipku za brisanje, postavljena je dodatna provjera koja je vidljiva na slici 4.6. Ona je u obliku skočnog prozora (engl. *pop-up window*) te sadrži dva odabira. Naravno, prvi je da se funkcionalnost izvrši, a drugi da se od navedenog odustane. Izlistanjem koda 4.6. prikazano je brisanje već spomenutih stavki.

```
private void ButtonRemoveTeam_Click(object sender, EventArgs e)
{
    string txtPath = Path.Combine("Teams Data", "Players", comboBoxAllTeams.Text);
    string imagePath = Path.Combine("Images", "Teams Logos", comboBoxAllTeams.Text);

    if (File.Exists(imagePath + ".jpg"))
        imagePath += ".jpg";
    if (File.Exists(imagePath + ".png"))
        imagePath += ".png";

    if (MessageBox.Show("Jeste li sigurni da želite obrisati ekipu " +
comboBoxAllTeams.Text + "?", "Upozorenje", MessageBoxButtons.YesNo,
MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        File.Delete(imagePath);
        File.Delete(txtPath + ".txt");
        DeleteTeamFromListOfTeams();
        MessageBox.Show("Ekipa uspješno obrisana", "Info", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        comboBoxAllTeams.Text = String.Empty;
        buttonRemoveTeam.Enabled = false;
        SetComboBox();
    }
}
```

Izlistanje koda 4.6. Metoda za klik na „Obriši“ ekipu

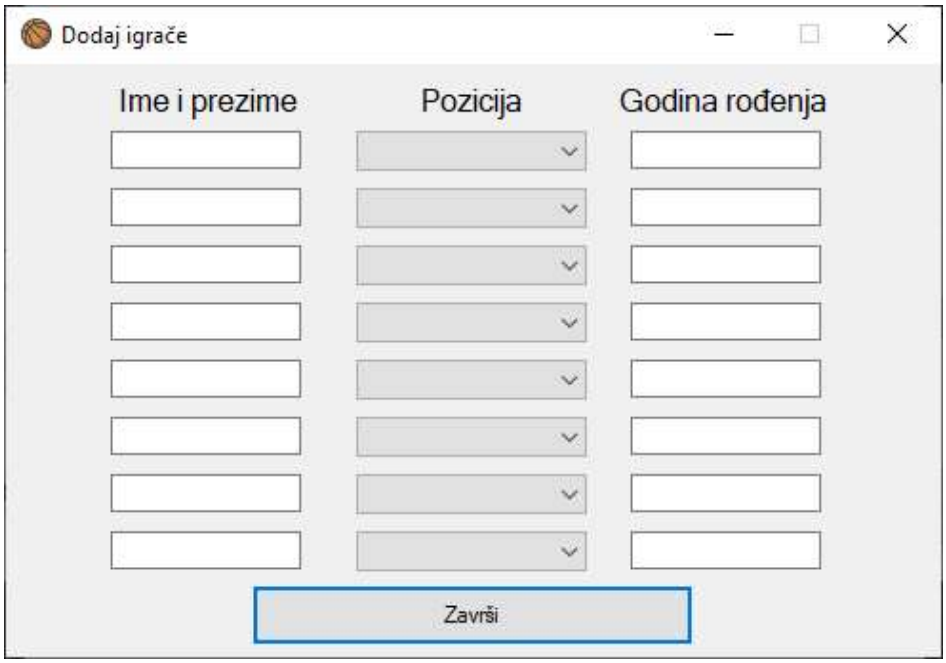
4.6. Forma za dodavanje igrača nove ekipe

Ovo potpoglavlje je već spomenuto u ranijem tekstu, a odnosilo se na broj igrača nove ekipe koji može biti proizvoljan, od osam do deset igrača. Budući da se taj broj mijenja od ekipe do ekipe, na ovu formu je bilo potrebno dinamički [12] dodavati kontrole. Navedena funkcionalnost prikazana je izlistanjem koda 4.7. Da bi se element dinamički dodao na formu, potrebno je stvoriti novi objekt željenog elementa. Nakon toga, potrebno mu je postaviti ime, dodati ga na željenu formu te postaviti lokaciju.

```
private static int yCoordinate = 35;
public static void MakeTextBoxes(Form addPlayersForm, int numberOfPlayers, int
labelLeftPoint, string status)
{
    for (int i = 0; i < numberOfPlayers; i++)
    {
        TextBox textBox = new TextBox();
        textBox.Name = "textBoxPlayer" + (i + 1).ToString() + status;
        addPlayersForm.Controls.Add(textBox);
        if (status == AddPlayersForm.STATUS_NAME)
            textBox.Location = new Point(labelLeftPoint, yCoordinate);
        if (status == AddPlayersForm.STATUS_YEAR_OF_BIRTH)
            textBox.Location = new Point(labelLeftPoint + 10, yCoordinate);
        yCoordinate += 30;
    }
    yCoordinate = 35;
}
```

Izlistanje koda 4.7. Dinamičko dodavanje elemenata na formu

To znači sljedeće, ako korisnik odabere osam igrača, na ovoj formi će se stvoriti osam elemenata za upisivanje imena igrača, osam elemenata za odabir pozicije igrača i osam elemenata za upisivanje godine rođenja igrača. Rezultat dinamičkog dodavanja elemenata prikazan je slikom 4.7.



Slika 4.7. Izgled forme za dodavanje igrača nove ekipe

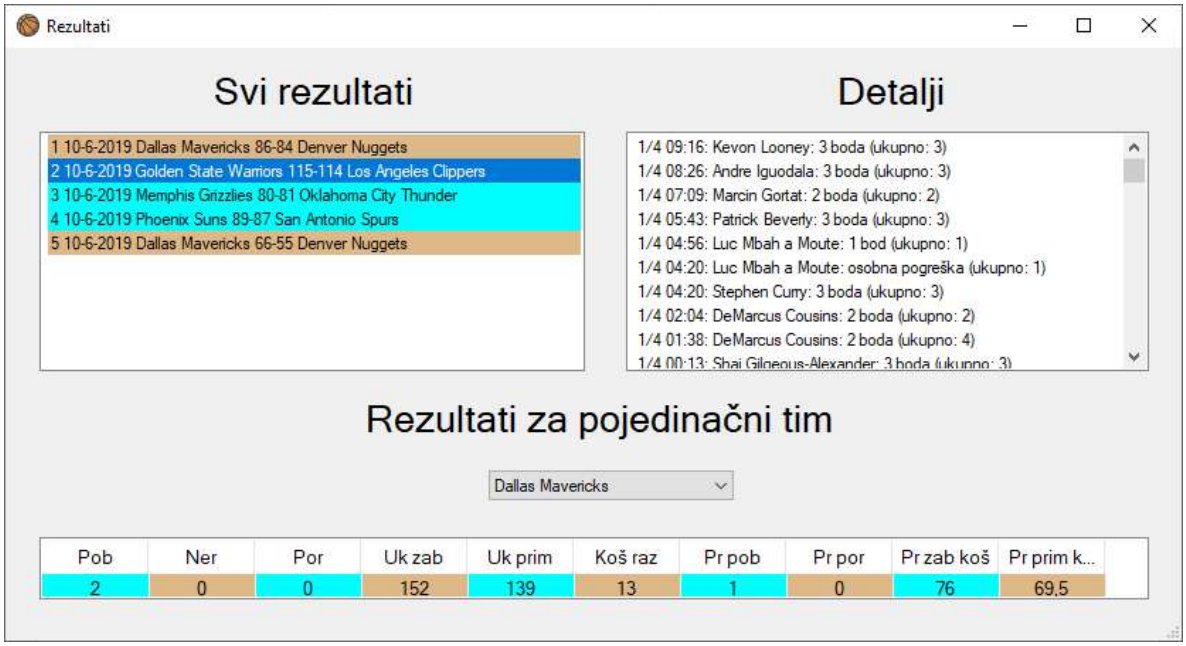
Kako se svaka nova ekipa i njeni igrači spremaju u datoteku te se isti učitavaju na pokretanju nove utakmice, bilo je potrebno provjeriti korisnikov unos za imena i godine. Kriteriji su bili da niti jedan element ne smije ostati neispunjen. Nadalje, imena ne smiju sadržavati brojke, prva znamenka godina ne smije biti nula te iste moraju biti u realnom rasponu. Za sprječavanje navedenog, korištene su vlastite klase iznimki (engl. *exception*). Primjer stvaranja iste te njezino upotrebljavanje prikazano je izlistanjem koda 4.8.

```
class BlankComboBoxException : Exception
{
    public BlankComboBoxException(string message) : base(message) { }
}


public static void CheckIfTextBoxesAreNotBlank(List<TextBox> textBoxes)
{
    foreach(TextBox tb in textBoxes)
    {
        if (tb.Text.Equals(String.Empty))
            throw new BlankTextBoxException("Unesite sve podatke");
    }
}
```

Izlistanje koda 4.8. Prikaz vlastite klase iznimke te njen poziv prilikom provjere

4.7.Forma za pregled rezultata

Za uspješno prikazivanje rezultata utakmica koje su se odigrale, bilo je potrebno iste spremiti, u ovom slučaju u datoteku. Rezultati se spremaju u jedinstvenu datoteku s ostalim rezultatima i to na način da se utakmica zapiše s brojem utakmice, koji predstavlja nekakvu vrstu ključa, kako ni jedna druga utakmica ne bi mogla biti zamijenjena s drugom. Ovo je bitno jer se prilikom odabira na redak u listi rezultata ispisuju detalji te utakmice.



Slika 4.8. Izgled forme za pregled rezultata

Na slici 4.8. je vidljivo da se pored broja utakmice nalazi i datum odigravanja, te na kraju i sam rezultat utakmice. Klikom na redak u listi rezultata, u drugom prozoru se ispisuju detalji utakmice. Oni podrazumijevaju svaki oblik utjecaja na utakmicu, a to uključuje koševe, prekršaje i prekide. Na kraju svakog retka se nalazi ukupan broj tog događaja, bio to koš, prekršaj ili prekid. Implementacija ove funkcionalnosti vidljiva je izlistanjem koda 4.9.

```
private void ListViewResults_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listViewResults.SelectedIndices.Count <= 0)
        return;
    else
        listViewDetails.Items.Clear();

    int index = listViewResults.SelectedIndices[0];
    if (index >= 0)
    {
        string text = listViewResults.SelectedItems[0].Text;
        string path = Path.Combine("Teams Data", "Games", "Statistics", text);
        List<string> list = MyFileReader.ReadFromFile(path + ".txt");
```

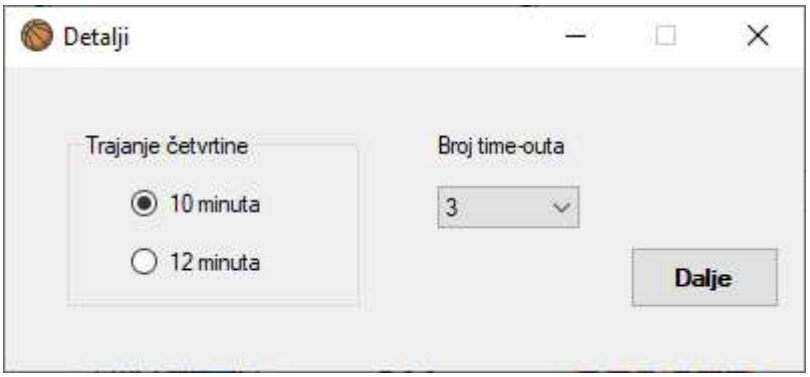
```
        foreach (string s in list)
        {
            listViewDetails.Items.Add(s);
        }
    }
}
```

Izlistanje koda 4.9. Postavljanje detalja utakmice za označenu utakmicu

Na ovoj formi je omogućeno i pretraživanje rezultata za pojedini tim. Potrebno je samo odabrati ekipu te se njena statistike ispisuje na za to odgovarajuće mjesto. Statistika obuhvaća broj pobjeda, poraza, neriješenih rezultata, zabijenih i primljenih koševa, koš razliku, prosječan broj pobjeda i poraza te prosječan broj zabijenih i primljenih koševa. Prilikom odabira ekipe kojoj želimo vidjeti statistiku, sve njene utakmice u gornjem prozoru su označene drugačijom bojom kako bi se lakše uočile.

4.8. Forma za postavljanje detalja utakmice

Budući da nije strogo definirano koliko utakmica treba imati prekida i koliko je trajanje jedne četvrtine, potrebno je omogućiti korisnikov izbor**.** Za navedeno je napravljena forma koja se sastoji od dva radijska gumba i jednog izbornika s padajućim izborom.

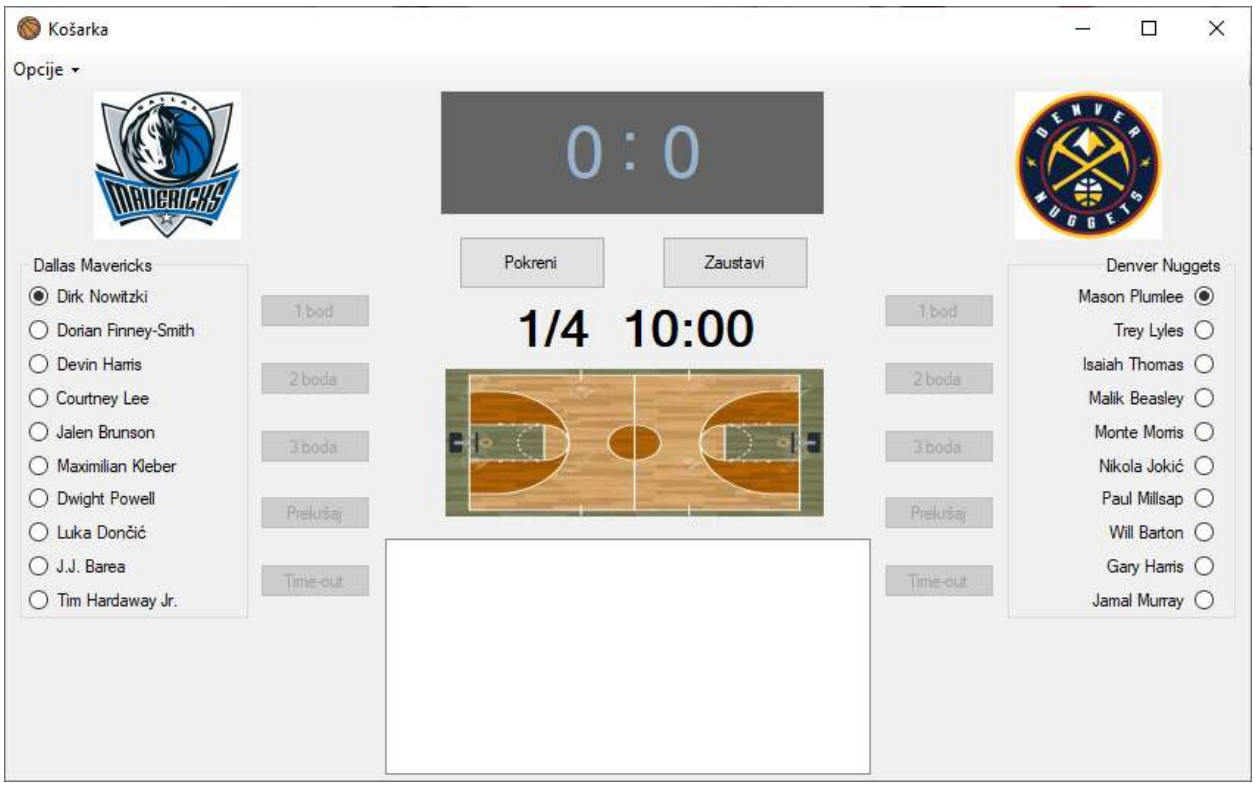


Slika 4.9. Izgled forme za postavljanje detalja

Kao što je vidljivo na slici 4.9., forma je poprilično jednostavna. Nakon klika na tipku „Dalje“ otvara se glavna forma u kojoj će se odvijati utakmica. Ista će biti opisana u idućem potpoglavlju.

4.9. Forma za tijek utakmice

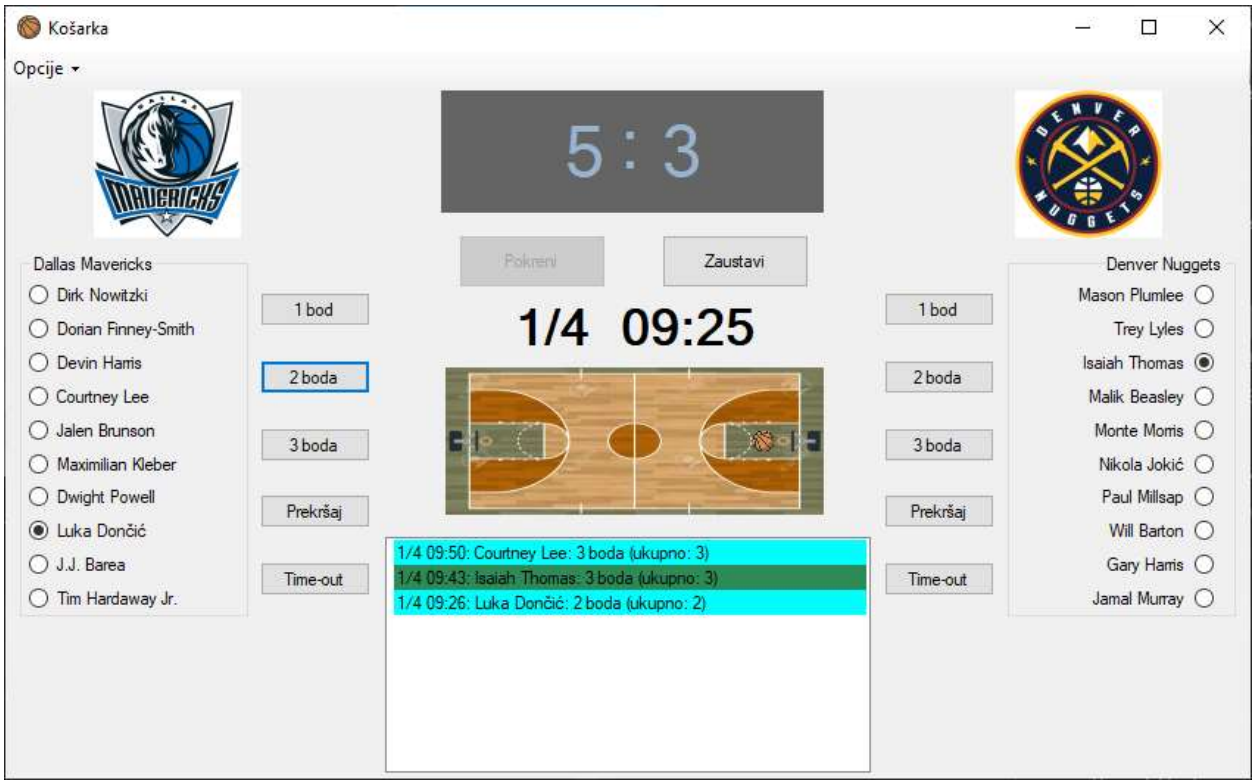
Glavni cilj ove aplikacije je, kao što naslov teme glasi, pratiti rezultat, odnosno tijek utakmice. Upravo to se odvija u ovom dijelu aplikacije. Kada korisnik postavi sve što je u prethodnim potpoglavljima opisano, otvara se forma prikazana na slici 4.10.



Slika 4.10. Izgled forme za tijek utakmice

Kao što je vidljivo na slici, s lijeve strane su detalji vezani za domaću ekipu, dok su na desnoj strani detalji za gostujuću. Svaka strana sadrži logo ekipe [13] te popis igrača [14]. U sredini se nalazi prikaz rezultata, tipke za pokretanje i zaustavljanje igre u slučaju prekida, slika košarkaškog igrališta [15] na kojemu se odvijaju animacije prilikom koševa ili prekida. Na dnu je prozor koji sprema zabilješke utakmice, odnosno detalje koji utječu na utakmicu.

Upravljanje ovom formom je poprilično jednostavno. Potrebno je samo odabrati igrača s popisa te kliknuti na jednu od tipki ovisno o tome koja se funkcionalnost želi obaviti. Nakon toga, dodaje se redak u listu detalja koji obavještava što se događa na utakmici.



Slika 4.11. Izgled forme za tijek utakmice s detaljima o postizanju koša

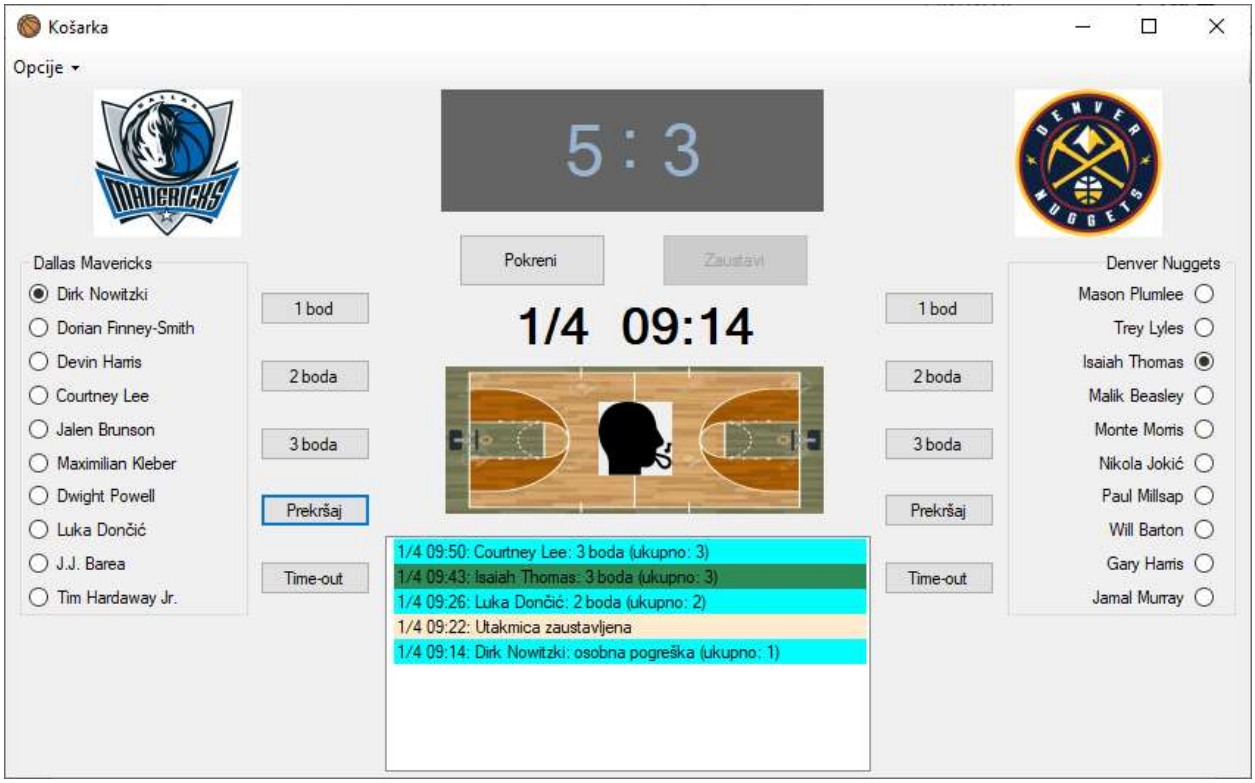
Redak koji prikazuje jedan detalj utakmice sadrži broj četvrtine, vrijeme kada se detalj dogodio te igrač koji je to učinio, što je i vidljivo na slici 4.11. Također, treba napomenuti da je svaki detalj koji predstavlja detalje za određenu ekipu prikazan drugačijom bojom radi boljeg pregleda. Prilikom postizanja koša, na slici igrališta se prikazuje lopta koja se kreće od mjesta ubačaja do koša. Mjesto ubačaja se može postaviti na tri različite lokacije, ovisno o tome je li to ubačaj za jedan, dva ili tri boda.

```
private void scorePointHomeAnimation()
{
    if (pictureBoxHome_Ball.Location.X != 520)
    {
        timerHome.Start();
        pictureBoxHome_Ball.Image = Properties.Resources.Ball;
        pictureBoxHome_Ball.Visible = true;
        int x = pictureBoxHome_Ball.Location.X;
        int newX = x + 1;
        pictureBoxHome_Ball.Location = new Point(newX, pictureBoxHome_Ball.Location.Y);
    }
    else
    {
        timerHome.Stop();
        pictureBoxHome_Ball.Visible = false;
    }
}
```

Izlistanje koda 4.10. Postavljanje animacije prilikom postizanja koša

Iz izlistanja koda 4.10. može se vidjeti kako animacija funkcionira. Prilikom klika na tipku za postizanje koša, lopta se postavlja na početnu lokaciju. Nakon toga se lopta svakih nekoliko milisekundi pomiče prema košu te se provjerava lokacija koja treba odgovarati lokaciji koša.

Na slici 4.12. prikazano je da se prilikom prekršaja prikazuje prigodna slika [16] te se korisnika obavještava zvučnim efektom [17]. U listu detalja se dodaje redak sličan onome za postizanje koša.



Slika 4.12. Izgled forme za tijek utakmice s detaljima o prekršajima

Nakon isteka posljednje sekunde zadnje četvrtine, svim kontrolama je onemogućena interakcija te je korisniku prikazan konačni rezultat i detalji o utakmici. Klikom na tipku opcije u statusnoj traci, otvara se padajući izbornik gdje korisnik može pokrenuti novu utakmicu, ispisati rezultat i detalje utakmice u datoteku te izaći iz aplikacije.

4.10. Nedostatci aplikacije i mogućnosti za nadogradnju

Svaka aplikacija, pa tako i ova, ima svoje mane i nedostatke. U ovome projektu postoji puno prostora za napredak. Međutim, budući da postoji rok za završetak istog, nije bilo mogućnosti za provedbu nadogradnje. Aplikacija, u ovim okvirima, sadrži sve prvobitno zamišljene funkcionalnosti te je pogodna za praćenje rezultata košarkaške utakmice.

Na prvi pogled, aplikacija bi mogla promijeniti dizajn da bi se poboljšao vizualni dojam. To bi uključivalo promjenu boje pozadine, kontrola i tako dalje. Što se tiče same funkcionalnosti aplikacije, bilo bi poželjno napraviti bazu podataka te ju koristiti umjesto datoteka u kojima su spremljeni podaci. Baza podataka može biti lokalna ili na internetu. Igrači već sada imaju statistiku, međutim, ona je poprilično siromašna ako ju usporedimo s nekim boljim rješenjima. Kako bi se to unaprijedilo, bilo bi potrebno pratiti i promašene prilike, a ne samo zabijene. Nadalje, skokovi, blokade i ukradene lopte su sastavni dio košarkaške utakmice te bi iste mogle biti implementirane u ovoj aplikaciji.

Na temelju znanja stečenog kroz izradu ove inačice aplikacije, navedene stavke za nadogradnju ne bi bile problem za napraviti.

5. ZAKLJUČAK

Živimo u vremenu kada se papir i olovka zamjenjuju računalima i mobitelima. To znači da se, kako u obrazovnom sustavu, tako i u sportskim aktivnostima sve više koriste moderne tehnologije koje značajno olakšavaju posao kojim se bavimo. Cilj ovog završnog rada bio je upravo ponuditi korisnicima jednostavno programsko rješenje za sportske aktivnosti, u ovom slučaju košarkašku utakmicu te olakšati način bilježenja utakmice.

Ovaj projekt je nastao iz ideje da svatko zainteresiran za košarku može napraviti svoju ekipu te kroz jednostavno sučelje pratiti rezultate iste. Izuzetno je pogodna za trenere ili neke amaterske ekipe koji žele imati pregled rezultata.

U izradi aplikacije bilo je potrebno proučiti veliku većinu korištenog znanja iz C# programskog jezika. Za to su korištene stavke navedene u literaturi. Upravo to znanje će uvelike pomoći u daljnjem razvoju iz programiranja te daljnjem unaprjeđenju aplikacije.

LITERATURA

[1] Kolegij Objektno orijentirano programiranje – Auditorne vježbe

[2] C. Watson, C. Nagel, J. Hammer Pedersen, J. D. Reid i M. Skinner, Beginning Visual C# 2010, Wiley Publishing, Inc., Inidanapolis, 2010

[3] Wikipedia: Microsoft Visual Studio, dostupno na: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio, pristupano: 6.5.2019.

[4] Easy Stats for Basketball, dostupno na: https://apps.apple.com/us/app/easy-stats-for-basketball/id561687907, pristupano: 10.5.2019.

[5] TurboStats, dostupno na: http://www.turbostats.com/newproducts.html, pristupano: 10.5.2019.

[6] Slika preuzeta s interneta, opis: animacija lopte, dostupno na: https://www.imagenesmy.com/imagenes/animated-basketball-goal-56.html, pristupano: 7.5.2019.

[7] Slika preuzeta s interneta, opis: animacija obavijesti korisniku, dostupno na: http://ajaxload.info/, pristupano: 7.5.2019.

[8] M. MacDonald, Pro .NET 2.0 Windows Forms and Custom Controls, Apress, New York, 2006

[9] Slika preuzeta s interneta, opis: košarkaška lopta, dostupno na: https://www.iconfinder.com/icons/2138358/ball_basketball_sport_team_sports_icon, pristupano: 7.5.2019.

[10] Microsoft Docs: File Class, dostupno na: https://docs.microsoft.com/en-us/dotnet/api/system.io.file?view=netframework-4.8, pristupano: 6.5.2019.

[11] C# Beginners Tutorial, dostupno na: https://www.youtube.com/watch?v=a1Ryv6DX6vU, pristupano: 7.5.2019.

[12] Microsoft Support: How to programmatically add controls to Windows forms at run time by using Visual C#, dostupno na: https://support.microsoft.com/en-us/help/319266/how-to-programmatically-add-controls-to-windows-forms-at-run-time-by-u, pristupano: 8.5.2019

[13] Slike preuzete s interneta, opis: logotipi zadanih ekipa, dostupno na: http://www.sportslogos.net, pristupano: 6.5.2019.

[14] Popis igrača preuzet s interneta, dostupno na: https://www.espn.com/nba/teams, pristupano: 6.5.2019.

[15] Slika preuzeta s interneta, opis: košarkaško igralište, dostupno na: https://www.123rf.com/photo_49537374_stock-vector-a-realistic-hardwood-textured-basketball-court-illustration-eps-10-file-contains-transparencies-.html, pristupano: 9.5.2019.

[16] Slika preuzeta s interneta, opis: sudac, dostupno na: https://www.visualpharm.com/free-icons/referee-595b40b75ba036ed117d8927, pristupano: 9.5.2019.

[17] Zvuk preuzet s interneta, opis: sudčev zvižduk, dostupno na: https://www.youtube.com/watch?v=CgTc_-A_Gzw, pristupano: 9.5.2019.

SAŽETAK

Cilj ovog završnog rada bio je izraditi aplikaciju s grafičkim sučeljem koja omogućuje praćenje rezultata i detalja košarkaške utakmice. U izradi iste, korišteno je razvojno okruženje Microsoft Visual Studio te C# programski jezik. Aplikacija se sastoji od više formi koje međusobno dijele podatke kako bi svaka imala mogućnost izvršavanja svog zadatka na ispravan način. Pri pokretanju aplikacije, korisnik odabire ekipe koje će igrati utakmicu, postavljaju se detalji te se otvara glavni prozor u kojemu se odvija tijek utakmice. Osim što omogućuje praćenje rezultata, korisniku je omogućeno i praćenje statistike za pojedinog igrača. Također, na formi se prikazuje lista događaja koji određuju utakmicu. Lista podrazumijeva spremanje zabijenih koševa, prekida te prekršaja uz koje je napisano vrijeme kada se dogodilo. Kako bi imao pregled odigranih utakmica, korisniku je omogućeno spremanje rezultata i detalja utakmice u datoteku koji se mogu prikazati u zasebnoj formi. Aplikacija nije ograničena na broj ekipa pa je omogućeno dinamičko dodavanje i uklanjanje istih.

KLJUČNE RIJEČI:

C#, košarka, Microsoft Visual Studio, rezultati, Windows Forms

ABSTRACT

Desktop application for keeping score of a basketball match

The goal of this final paper was to create an application with a graphical interface which allows keeping score and details of a basketball match. For its development, Microsoft Visual Studio and C# programming language were used. The application consists of multiple forms which share data to each other so that everyone has the ability to perform their task in the correct manner. When launching the application, a user selects teams to play the game, sets up the details and opens the main window where the match takes place. In addition to tracking results, the user has ability to track statistics for each player. Also, the form shows the event list that determines the match. The list implies storing scored points, breaks and fouls with the written time when it happened. In order to have an overview of the played games, the user is allowed to save the result and the match details to a file that can be displayed in a separate form. The application is not limited to the number of teams so dynamically adding and removing them is possible.

KEYWORDS:

basketball, C#, Microsoft Visual Studio, results, Windows Forms

ŽIVOTOPIS

Matija Sokol rođen je 3. rujna 1997. godine u Našicama. Osnovnu školu pohađao je i završio u Našicama u školi kralja Tomislava u razdoblju od 2004. do 2012. godine. Zatim upisuje prirodoslovno matematičku gimnaziju u Srednjoj školi Isidora Kršnjavoga u Našicama koju pohađa od 2012. do 2016. godine. Nakon završene srednje škole upisuje preddiplomski studij računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku 2016. godine. Tijekom treće godine preddiplomskog studija pohađa i uspješno završava Android Dev Academy u organizaciji Osijek Software City-a.

Vlastoručni potpis:

_____

Matija Sokol