

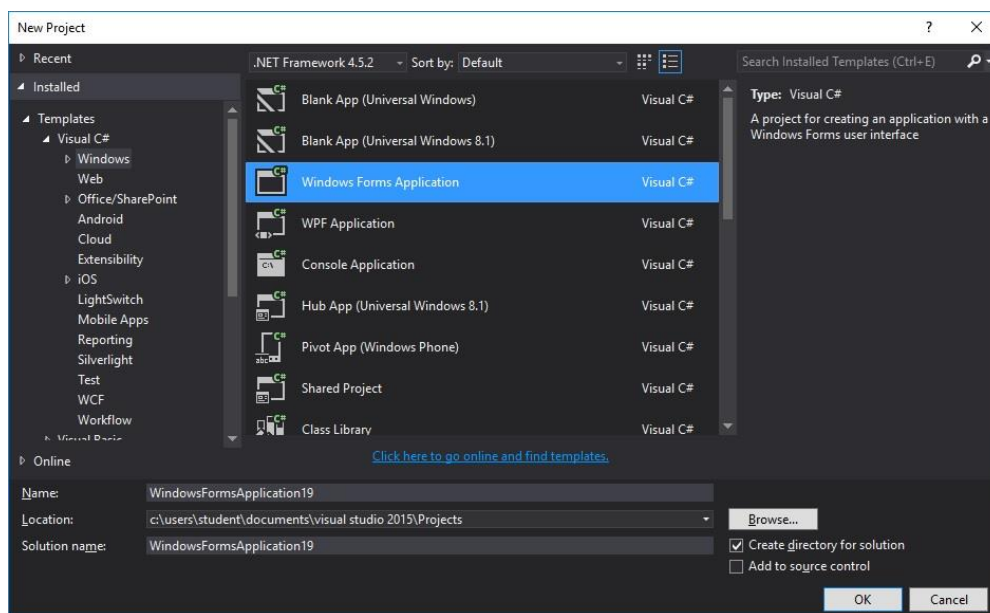
## DELEGATI

Zadatak je napraviti Windows aplikaciju male knjižnice.

Knjižnica mora imati sljedeća obilježja : potrebno je omogućiti korisniku prijavu u aplikaciju.

Nakon provjerene prijave korisniku će biti prikazana ili poruka o greški ili popis knjiga.

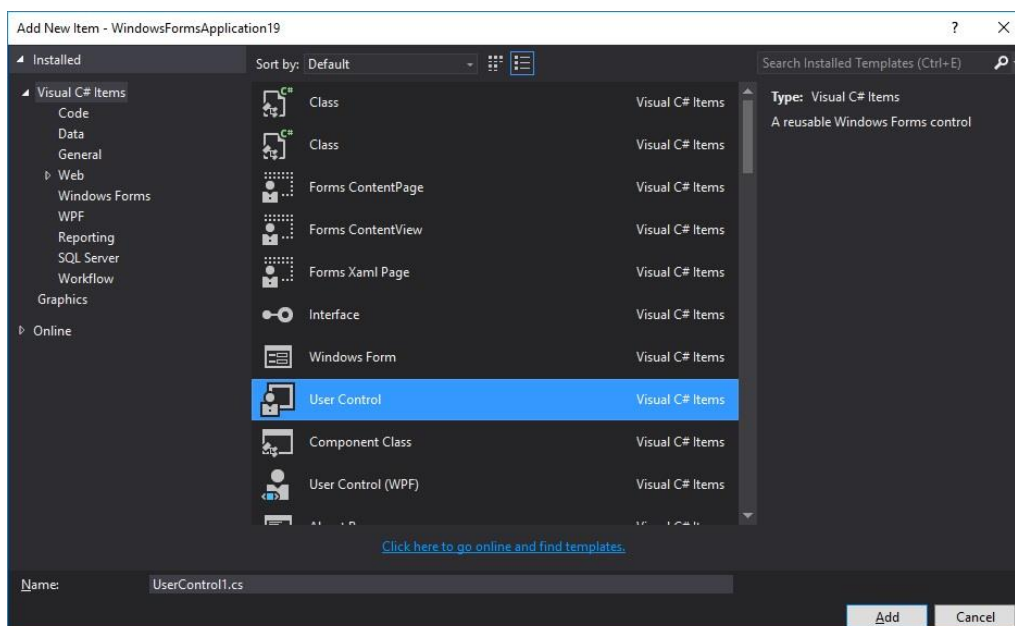
U Visual Studiu stvorite novu Windows aplikaciju.



Knjižnica mora biti izgrađena od jedne forme i četiri kontrole (prijava, pregled, greska i kraj).

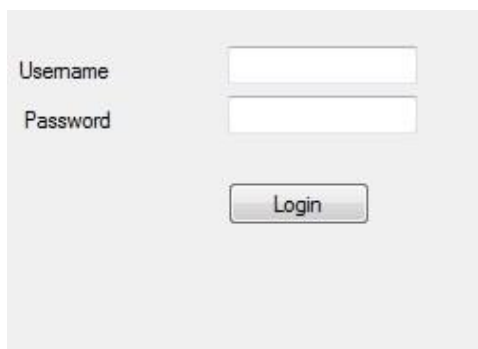
Glavna forma će na sebi sadržavati sve kontrole koje će biti prikazivane odnosno skrivane postavljanjem njihova „Visible“ stanja na vrijednosti istina(true) ili laž(false).

Dodajte novu kontrolu naziva prijava:



Novostvorenoj kontroli potrebno je dati sve željene elemente: dva tekstualna polja (textBox), dvije labele i jedan gumb. Ona bi trebala izgledati približno:

LogIn.cs:

A screenshot of a login form. It has a light gray background. On the left, there are two labels: 'Username' and 'Password'. To the right of 'Username' is a white text box. To the right of 'Password' is another white text box. Below these two text boxes is a button labeled 'Login'.

Na isti način kreirajte i ostale kontrole čiji izgled neka prati sljedeće modele:

koji kao glavni sastavni element ima *dataGridView*

PregledKnjiga.cs:

A screenshot of a book review form. It has a light beige background. In the center, there is a large, dark gray rectangular area representing a data grid. Below this grid, on the left side, is a button labeled 'End'.

Greska.cs:

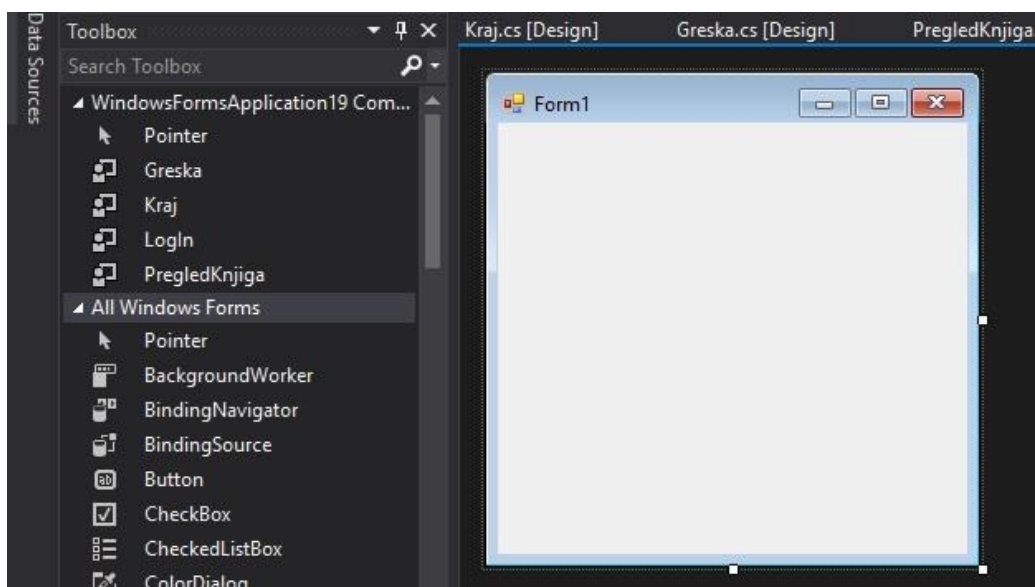
A screenshot of an error dialog box. It has a light gray background. In the center, the text 'User not found' is displayed in a yellow-green color. Below the text is a button labeled 'OK'.

Kraj.cs:

Thank you for using library

OK

Kada ste sve kontrole izradili, promjene je potrebno pospremiti. U vašoj alatnoj traci će tada doći do izmjena jer će biti dodate novostvorene kontrole koje je moguće „dovući“ na formu:



Knjižnica će sve potrebne podatke imati pohranjene u XML datotekama. Postupak je kao i u prošloj vježbi. Za spajanje XML datoteka i programkog koda koristite sljedeće:

```
private XDocument korisnici;
```

```
.  
. .
```

```
korisnici = XDocument.Load("korisnici.xml");
```

```
.  
. .
```

```
var k = from x in korisnici.Descendants("korisnik")
```

```
select new
```

```
{  
    korisnickoIme = x.Descendants("korisnickoIme").First().Value,
```

```
lozinka = x.Descendants("lozinka").First().Value,  
};
```

## DOGAĐAJI

Svaka od dodanih kontrola mora biti u mogućnosti glavnoj formi dojaviti da se desio određeni događaj kako bi ona mogla reagirati (neku kontrolu prikazati a neku sakriti).

Komunikaciju među njima implementirat ćemo korištenjem događaja.

Na samom startu naša aplikacija ima skrivene sve kontrole osim prijave. Kontrola prijava bi na pritisak gumba trebala provjeriti da li u bazi postoji korisnik sa unesenim podacima. Ako on postoji treba dojaviti formi da je sve prošlo u redu u protivnom potrebno joj je javiti da se desila greška.

Da bi se mogli javljati događaji u C# je potrebno definirati delegata, a zatim događaj koji koristi potpis tog delegata. U našoj kontroli prijava delegat sa pripadajućim događajem definiran je:

```
public delegate void LoginHandler(string username);  
public event LoginHandler LoginSuccessful;
```

tada se taj događaj u željenom trenutku može objaviti na jednostavan način:

```
LoginSuccessful("User found");
```

Naš objavljeni događaj netko mora i obraditi. U ovom primjeru glavna forma je ta koja bi trebala reagirati i na objavu događaja, u ovom slučaju na događaj da je korisnik uspješno prijavljen. Tada je njezin zadatak kontrolu prijava sakriti a pregled prikazati. Da bi bila u mogućnosti dobivati obavijesti o određenom događaju kontrola se mora za njega prijaviti:

```
public Form1()  
{  
    InitializeComponent();  
    Subscribe();  
}  
  
public void Subscribe()  
{  
    logIn1.LoginSuccessful += new LoginHandler(logIn1_Successful);  
    .  
    .  
    .  
}
```

U prijavi koja se dešava u metodi `Subscribe` potrebno je navesti i ime metode, `logIn1_Successful`, koja će određeni događaj obraditi a čiji potpis u potpunosti mora odgovarati definiranom delegatu

Na ovaj način ugradite komunikaciju svih elemenata na formi.