Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

**Objektno oblikovanje**

*Projektni seminar za „Sustav za programsku podršku knjigovodstvu udruga“*

Matea Cerovac (0036458952)

Juraj Dončević (0036463062)

Matej Vukosav (0036473765)

Zagreb, 16.1.2017.

Sadržaj

[1. Design story 3](#_Toc473414102)

[2. Use cases (slučajevi korištenja) 5](#_Toc473414103)

[3. Backend sustava 17](#_Toc473414104)

[4. Mockup zasloni 18](#_Toc473414105)

[4.1 Desktop aplikacija 18](#_Toc473414106)

[4.1.1 Glavni zaslon 18](#_Toc473414107)

[4.1.2 Zaslon generiranja izvješća 19](#_Toc473414108)

[4.1.3 Zasloni pregleda i uređivanja podataka 19](#_Toc473414109)

[4.2 Web aplikacija 20](#_Toc473414110)

[4.2.1 Glavni zaslon 20](#_Toc473414111)

[4.2.2 Zaslon generiranja izvješća 21](#_Toc473414112)

[4.2.3 Zaslon pregleda podataka 22](#_Toc473414113)

[4.2.4 Zaslon za unos i uređivanje podataka 22](#_Toc473414114)

[5. Opis implementirane web aplikacije 24](#_Toc473414115)

[5.1 Zaslon prijave na sustav 24](#_Toc473414116)

[5.2 Izbornik 24](#_Toc473414117)

[5.3 Pregled knjigovodstvene knjige 25](#_Toc473414118)

[5.4 Dodavanje nove stavke 26](#_Toc473414119)

[5.5 Uređivanje postojeće stavke 27](#_Toc473414120)

[5.6 Izvješća 28](#_Toc473414121)

[5.7 Opis korištene MVC arhitekture 28](#_Toc473414122)

[6. Opis implementirane Desktop aplikacije 30](#_Toc473414123)

[6.1 Zaslon prijave na sustav 30](#_Toc473414124)

[6.2 Izbornik 30](#_Toc473414125)

[6.3 Dodavanje nove stavke 32](#_Toc473414126)

[6.4 Ažuriranje podataka u postojećoj stavci 33](#_Toc473414127)

[6.5 Izvješća 34](#_Toc473414128)

[6.6 Opis iskorištene MVVM arhitekture 35](#_Toc473414129)

# Design story

1. siječnja 2015. godine na snagu je stupio novi Zakon o financijskom poslovanju i računovodstvu neprofitnih organizacija (NN 121/2014). Po zakonu neprofitne su one organizacije koje 3 godine uzastopno nisu imale financijsku dobit veću od 230 000 kn i 3 godine uzastopno nisu posjedovale ukupnu imovinu u višoj vrijednosti od 230 000 kn. U osnovi veliki broj udruga su neprofitne organizacije, jer ne premašuju financijska ograničenja zadana zakonom. Zakon propisuje svim neprofitnim organizacijama vođenje *jednostavnog knjigovodstva*, te podnošenje godišnjih financijskih izvješća.

Većina udruga se našla pred problemom, jer do tada nisu bile obavezne voditi nikakvo knjigovodstvo. Problem je još time veći što se udruge baziraju na ideji volonterizma, gdje članovi udruge „doniraju“ svoje vrijeme i rad, ali ne moraju biti stručni u svim vidovima vođenja udruge. Tako su se udruge našle pred nedostatkom stručnog kadra za vođenje knjigovodstva, a oni članovi koji su prihvatili zadatak vođenja knjigovodstva su ga većinom prihvatili nevoljko. Ideja je stvoriti programsku podršku koja će pojednostaviti članovima udruga vođenje knjigovodstva.

Korisnici bi u sklopu programske podrške trebali moći upravljati svojim knjigovodstvenim knjigama, te im treba biti omogućeno stvaranje izvještaja (među kojima je i godišnje financijsko izvješće) koji se baziraju na knjigovodstvenim knjigama. Knjige kojima bi pohrana i uređivanje bilo podržano su: knjige primitaka i izdataka, te knjige ulaznih i izlaznih računa.

Knjige primitaka i izdataka drže podatke o svim novčanim primicima i izdacima koje je udruga ostvarila. Primici i izdaci mogu biti u obliku gotovinske transakcije, transakcije putem žiro-računa i transakcije u naravi (plaćanje u obliku nekog djela ili usluge). Dobra napomena je da se primitak i izdatak ne mora odviti u isključivo jednom obliku.

Knjige izlaznih i ulaznih računa su popisi svih dokumenata naplate i isplate koje je udruga napravila. To je popis svih računa i uplatnica koje je udruga izdala (takvi podaci su dio knjige izlaznih računa), te popis svih dokumenata naplate koje je udruga zaprimila (podaci knjige ulaznih računa).

U osnovi ideja takvog knjigovodstva je da se putem ulaznih i izlaznih računa definiraju sve službene poslovne transakcije, dok se knjigama primitaka i izdataka pravdaju uplate i isplate koje udruga napravi.

Postoje pokušaji implementacije vlastite programske podrške od strane članova raznih udruga, no ona se često svode na primjenu Excel i Access programskih alata. Problem je u tome što se podaci pohranjuju samo lokalno, te postoji ovisnost o postojanju *Microsoft Office* alata na lokalnom računalnom sustavu. Dakle, programska potpora bi trebala biti pristupačna sa većine računalnih sustava, uz minimalnu potrebu vanjskih alata. U krajnjem slučaju instalacija programske podrške na lokalnom sustavu bi trebala biti dostatna. Također postoji i problem pohranjivanja, jer članovi drže podatke na svojim lokalnim skladištima podataka. To bi trebalo biti riješeno centraliziranim pristupom podacima, odnosno uvodi se jedinstvena baza podataka. Time se članovi ne bi trebali brinuti o ažurnosti svoje lokalne kopije knjigovodstvenih podataka.

Optimalno rješenje bi bila implementacija desktop i web aplikacije koje će raditi nad jedinstvenom bazom podataka, te će korisnicima (članovima udruga) omogućiti pregledno vođenje knjigovodstvenih podataka svoje udruge.

# Use cases (slučajevi korištenja)

Za sustav definirani su slijedeći slučajevi korištenja

1. Prijava korisnika na sustav
2. Pregled knjige primitaka
3. Pregled knjige izdataka
4. Pregled knjige izlaznih računa
5. Pregled knjige ulaznih računa
6. Dodavanje stavke u knjigu primitaka
7. Dodavanje stavke u knjigu izdataka
8. Dodavanje stavke u knjigu izlaznih računa
9. Dodavanje stavke u knjigu ulaznih računa
10. Uređivanje stavke u knjizi primitaka
11. Uređivanje stavke u knjizi izdataka
12. Uređivanje stavke u knjizi izlaznih računa
13. Uređivanje stavke u knjizi ulaznih računa
14. Brisanje stavke u knjizi primitaka
15. Brisanje stavke u knjizi izdataka
16. Brisanje stavke u knjizi izlaznih računa
17. Brisanje stavke u knjizi ulaznih računa
18. Stvaranje godišnjeg financijskog izvješća
19. Stvaranje izvješća toka novca

|  |
| --- |
| **Naziv slučaja korištenja:** Prijava korisnika na sustav |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi pristupiti značajkama sustava  Sustav – omogućava pristup knjigovodstvenim podacima |
| **Preduvjeti:**   1. Korisnik je registriran u sustavu |
| **Post-uvjeti:**   1. Uspješno prijavljivanje korisnika na sustav |
| **Glavni uspješni scenarij:**   1. Korisnik pokreće aplikaciju 2. Aplikacija se uspješno pokreće i traži korisnika korisničko ime i lozinku 3. Korisnik upisuje korisničko ime i lozinku 4. Aplikacija provjerava postojanje korisnika i ispravnost lozinke 5. Aplikacija dopušta korisniku funkcionalnosti i prikazuje glavni zaslon |
| **Alternativni scenarij:**  2a. Aplikacija nije uspješno pokrenuta zbog programske pogreške  2a1. Aplikacija ispisuje poruku o pogreškama  5a. Aplikacija ne pronalazi navedeno korisničko ime i lozinku u sustavu  5a1. Aplikacija odbija korisniku pristup funkcionalnostima  5a2. Aplikacija omogućava ponovni unos korisničkog imena i lozinke |

|  |
| --- |
| **Naziv slučaja korištenja:** Pregled knjige primitaka |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi pristupiti pregledu knjige primitaka  Sustav – omogućava pristup podacima iz knjige primitaka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustav |
| **Post-uvjeti:**   1. Korisniku je predočen popis stavki u pripadajućoj knjizi primitaka |
| **Glavni uspješni scenarij:**   1. Korisniku se prikazuju stavke knjige primitaka koji pripadaju njegovoj udruzi 2. Korisniku je dopušten pristup uređivanju, brisanju i stvaranju stavki |
| **Alternativni scenarij:**  2a. Korisnik ne posjeduje stavke u knjizi primitaka  2a1. Korisniku je omogućeno samo stvaranje nove stavke |

|  |
| --- |
| **Naziv slučaja korištenja:** Pregled knjige izdataka |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi pristupiti pregledu knjige izdataka  Sustav – omogućava pristup podacima iz knjige izdataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustav |
| **Post-uvjeti:**   1. Korisniku je predočen popis stavki u pripadajućoj knjizi izdataka |
| **Glavni uspješni scenarij:**   1. Korisniku se prikazuju stavke knjige izdataka koji pripadaju njegovoj udruzi 2. Korisniku je dopušten pristup uređivanju, brisanju i stvaranju stavki |
| **Alternativni scenarij:**  2a. Korisnik ne posjeduje stavke u knjizi izdataka  2a1. Korisniku je omogućeno samo stvaranje nove stavke |

|  |
| --- |
| **Naziv slučaja korištenja:** Pregled knjige izlaznih računa |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi pristupiti pregledu knjige izlaznih računa  Sustav – omogućava pristup podacima iz knjige izlaznih računa |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustav |
| **Post-uvjeti:**   1. Korisniku je predočen popis stavki u pripadajućoj knjizi izlaznih računa |
| **Glavni uspješni scenarij:**   1. Korisniku se prikazuju stavke knjige izlaznih računa koji pripadaju njegovoj udruzi 2. Korisniku je dopušten pristup uređivanju, brisanju i stvaranju stavki |
| **Alternativni scenarij:**  2a. Korisnik ne posjeduje stavke u knjizi izlaznih računa  2a1. Korisniku je omogućeno samo stvaranje nove stavke |

|  |
| --- |
| **Naziv slučaja korištenja:** Pregled knjige ulaznih računa |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi pristupiti pregledu knjige ulaznih računa  Sustav – omogućava pristup podacima iz knjige ulaznih računa |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustav |
| **Post-uvjeti:**   1. Korisniku je predočen popis stavki u pripadajućoj knjizi ulaznih računa |
| **Glavni uspješni scenarij:**   1. Korisniku se prikazuju stavke knjige ulaznih računa koji pripadaju njegovoj udruzi 2. Korisniku je dopušten pristup uređivanju, brisanju i stvaranju stavki |
| **Alternativni scenarij:**  2a. Korisnik ne posjeduje stavke u knjizi ulaznih računa  2a1. Korisniku je omogućeno samo stvaranje nove stavke |

|  |
| --- |
| **Naziv slučaja korištenja:** Dodavanje stavke u knjigu primitaka |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi dodati novu stavku u knjigu primitaka  Sustav – omogućava spremanje i validaciju novih podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Uspješno stvorena nova stavka knjige primitaka |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire stvaranje nove stavke 2. Sustav prikazuje formu za unos atributa stavke 3. Korisnik popunjava polja atributa stavke primitka 4. Korisnik zahtjeva pohranu podataka 5. Sustav uspješno validira unesene vrijednosti 6. Sustav uspješno pohranjuje podatke 7. Korisnik se vraća na pregled knjige primitaka |
| **Alternativni scenarij:**  5a. Sustav neuspješno validira neispravno unesene vrijednosti  5a1. Sustav korisniku dojavljuje neispravno unesene vrijednosti  5a2. Korisniku je onemogućeno spremanje podataka sve dok ne prođu validaciju  5a3. Korisnik je unesao ispravne vrijednosti i zahtjeva pohranu podataka  5a4. Sustav sprema podatke i vraća korisnika na pregled knjige primitaka  6a. Sustav zbog pogreške ne uspijeva spremiti podatke  6a1. Sustav odustaje od promjena  6a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Dodavanje stavke u knjigu izdataka |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi dodati novu stavku u knjigu izdataka  Sustav – omogućava spremanje i validaciju novih podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Uspješno stvorena nova stavka knjige izdataka |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire stvaranje nove stavke 2. Sustav prikazuje formu za unos atributa stavke 3. Korisnik popunjava polja atributa stavke izdatka 4. Korisnik zahtjeva pohranu podataka 5. Sustav uspješno validira unesene vrijednosti 6. Sustav uspješno pohranjuje podatke 7. Korisnik se vraća na pregled knjige izdataka |
| **Alternativni scenarij:**  5a. Sustav neuspješno validira neispravno unesene vrijednosti  5a1. Sustav korisniku dojavljuje neispravno unesene vrijednosti  5a2. Korisniku je onemogućeno spremanje podataka sve dok ne prođu validaciju  5a3. Korisnik je unesao ispravne vrijednosti i zahtjeva pohranu podataka  5a4. Sustav sprema podatke i vraća korisnika na pregled knjige izdataka  6a. Sustav zbog pogreške ne uspijeva spremiti podatke  6a1. Sustav odustaje od promjena  6a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Dodavanje stavke u knjigu izlaznih računa |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi dodati novu stavku u knjigu izlaznih računa  Sustav – omogućava spremanje i validaciju novih podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Uspješno stvorena nova stavka knjige izlaznih računa |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire stvaranje nove stavke 2. Sustav prikazuje formu za unos atributa stavke 3. Korisnik popunjava polja atributa stavke izlaznog računa 4. Korisnik zahtjeva pohranu podataka 5. Sustav uspješno validira unesene vrijednosti 6. Sustav uspješno pohranjuje podatke 7. Korisnik se vraća na pregled knjige izlaznih računa |
| **Alternativni scenarij:**  5a. Sustav neuspješno validira neispravno unesene vrijednosti  5a1. Sustav korisniku dojavljuje neispravno unesene vrijednosti  5a2. Korisniku je onemogućeno spremanje podataka sve dok ne prođu validaciju  5a3. Korisnik je unesao ispravne vrijednosti i zahtjeva pohranu podataka  5a4. Sustav sprema podatke i vraća korisnika na pregled knjige izlaznih računa  6a. Sustav zbog pogreške ne uspijeva spremiti podatke  6a1. Sustav odustaje od promjena  6a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Dodavanje stavke u knjigu ulaznih računa |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi dodati novu stavku u knjigu ulaznih računa  Sustav – omogućava spremanje i validaciju novih podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Uspješno stvorena nova stavka knjige ulaznih računa |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire stvaranje nove stavke 2. Sustav prikazuje formu za unos atributa stavke 3. Korisnik popunjava polja atributa stavke ulaznog računa 4. Korisnik zahtjeva pohranu podataka 5. Sustav uspješno validira unesene vrijednosti 6. Sustav uspješno pohranjuje podatke 7. Korisnik se vraća na pregled knjige ulaznih računa |
| **Alternativni scenarij:**  5a. Sustav neuspješno validira neispravno unesene vrijednosti  5a1. Sustav korisniku dojavljuje neispravno unesene vrijednosti  5a2. Korisniku je onemogućeno spremanje podataka sve dok ne prođu validaciju  5a3. Korisnik je unesao ispravne vrijednosti i zahtjeva pohranu podataka  5a4. Sustav sprema podatke i vraća korisnika na pregled knjige ulaznih računa  6a. Sustav zbog pogreške ne uspijeva spremiti podatke  6a1. Sustav odustaje od promjena  6a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Uređivanje stavke u knjizi primitaka |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi urediti stavku u knjizi primitaka  Sustav – omogućava spremanje i validaciju izmijenjenih podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Uspješno spremljena izmijenjena stavka |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire onu koju želi izmijeniti 2. Sustav prikazuje formu za izmjenu atributa stavke 3. Korisnik izmjenjuje polja atributa stavke primitka 4. Korisnik zahtjeva pohranu podataka 5. Sustav uspješno validira unesene vrijednosti 6. Sustav uspješno pohranjuje podatke 7. Korisnik se vraća na pregled knjige primitaka |
| **Alternativni scenarij:**  5a. Sustav neuspješno validira neispravno unesene vrijednosti  5a1. Sustav korisniku dojavljuje neispravno unesene vrijednosti  5a2. Korisniku je onemogućeno spremanje podataka sve dok ne prođu validaciju  5a3. Korisnik je unesao ispravne vrijednosti i zahtjeva pohranu podataka  5a4. Sustav sprema podatke i vraća korisnika na pregled knjige primitaka  6a. Sustav zbog pogreške ne uspijeva spremiti podatke  6a1. Sustav odustaje od promjena  6a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Uređivanje stavke u knjizi izdataka |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi urediti stavku u knjizi izdataka  Sustav – omogućava spremanje i validaciju izmijenjenih podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Uspješno spremljena izmijenjena stavka |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire onu koju želi izmijeniti 2. Sustav prikazuje formu za izmjenu atributa stavke 3. Korisnik izmjenjuje polja atributa stavke izdatka 4. Korisnik zahtjeva pohranu podataka 5. Sustav uspješno validira unesene vrijednosti 6. Sustav uspješno pohranjuje podatke 7. Korisnik se vraća na pregled knjige izdataka |
| **Alternativni scenarij:**  5a. Sustav neuspješno validira neispravno unesene vrijednosti  5a1. Sustav korisniku dojavljuje neispravno unesene vrijednosti  5a2. Korisniku je onemogućeno spremanje podataka sve dok ne prođu validaciju  5a3. Korisnik je unesao ispravne vrijednosti i zahtjeva pohranu podataka  5a4. Sustav sprema podatke i vraća korisnika na pregled knjige izdataka  6a. Sustav zbog pogreške ne uspijeva spremiti podatke  6a1. Sustav odustaje od promjena  6a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Uređivanje stavke u knjizi izlaznih računa |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi urediti stavku u knjizi izlaznih računa  Sustav – omogućava spremanje i validaciju izmijenjenih podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Uspješno spremljena izmijenjena stavka |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire onu koju želi izmijeniti 2. Sustav prikazuje formu za izmjenu atributa stavke 3. Korisnik izmjenjuje polja atributa stavke izlaznog računa 4. Korisnik zahtjeva pohranu podataka 5. Sustav uspješno validira unesene vrijednosti 6. Sustav uspješno pohranjuje podatke 7. Korisnik se vraća na pregled knjige izlaznih računa |
| **Alternativni scenarij:**  5a. Sustav neuspješno validira neispravno unesene vrijednosti  5a1. Sustav korisniku dojavljuje neispravno unesene vrijednosti  5a2. Korisniku je onemogućeno spremanje podataka sve dok ne prođu validaciju  5a3. Korisnik je unesao ispravne vrijednosti i zahtjeva pohranu podataka  5a4. Sustav sprema podatke i vraća korisnika na pregled knjige izlaznih računa  6a. Sustav zbog pogreške ne uspijeva spremiti podatke  6a1. Sustav odustaje od promjena  6a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Uređivanje stavke u knjizi ulaznih računa |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi urediti stavku u knjizi ulaznih računa  Sustav – omogućava spremanje i validaciju izmijenjenih podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Uspješno spremljena izmijenjena stavka |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire onu koju želi izmijeniti 2. Sustav prikazuje formu za izmjenu atributa stavke 3. Korisnik izmjenjuje polja atributa stavke ulaznog računa 4. Korisnik zahtjeva pohranu podataka 5. Sustav uspješno validira unesene vrijednosti 6. Sustav uspješno pohranjuje podatke 7. Korisnik se vraća na pregled knjige ulaznih računa |
| **Alternativni scenarij:**  5a. Sustav neuspješno validira neispravno unesene vrijednosti  5a1. Sustav korisniku dojavljuje neispravno unesene vrijednosti  5a2. Korisniku je onemogućeno spremanje podataka sve dok ne prođu validaciju  5a3. Korisnik je unesao ispravne vrijednosti i zahtjeva pohranu podataka  5a4. Sustav sprema podatke i vraća korisnika na pregled knjige ulaznih računa  6a. Sustav zbog pogreške ne uspijeva spremiti podatke  6a1. Sustav odustaje od promjena  6a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Brisanje stavke u knjizi primitaka |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi izbrisati stavku iz knjige primitaka  Sustav – omogućava brisanje podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Stavka je uspješno izbrisana iz sustava |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire onu koju želi obrisati 2. Sustav prikazuje prozor sigurnosnog upita 3. Korisnik potvrđuje brisanje 4. Sustav uspješno briše podatke 5. Korisnik se vraća na pregled knjige primitaka |
| **Alternativni scenarij:**  3a. Korisnik odbacuje postupak brisanja stavke  3a1. Sustav ne provodi brisanje i korisnika vraća na pregled stavki  4a. Sustav zbog pogreške ne uspijeva pohraniti promjene  4a1. Sustav odustaje od promjena  4a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Brisanje stavke u knjizi izdataka |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi izbrisati stavku iz knjige izdataka  Sustav – omogućava brisanje podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Stavka je uspješno izbrisana iz sustava |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire onu koju želi obrisati 2. Sustav prikazuje prozor sigurnosnog upita 3. Korisnik potvrđuje brisanje 4. Sustav uspješno briše podatke 5. Korisnik se vraća na pregled knjige izdataka |
| **Alternativni scenarij:**  3a. Korisnik odbacuje postupak brisanja stavke  3a1. Sustav ne provodi brisanje i korisnika vraća na pregled stavki  4a. Sustav zbog pogreške ne uspijeva pohraniti promjene  4a1. Sustav odustaje od promjena  4a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Brisanje stavke u knjizi izlaznih računa |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi izbrisati stavku iz knjige izlaznih računa  Sustav – omogućava brisanje podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Stavka je uspješno izbrisana iz sustava |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire onu koju želi obrisati 2. Sustav prikazuje prozor sigurnosnog upita 3. Korisnik potvrđuje brisanje 4. Sustav uspješno briše podatke 5. Korisnik se vraća na pregled knjige izlaznih računa |
| **Alternativni scenarij:**  3a. Korisnik odbacuje postupak brisanja stavke  3a1. Sustav ne provodi brisanje i korisnika vraća na pregled stavki  4a. Sustav zbog pogreške ne uspijeva pohraniti promjene  4a1. Sustav odustaje od promjena  4a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Brisanje stavke u knjizi ulaznih računa |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi izbrisati stavku iz knjige ulaznih računa  Sustav – omogućava brisanje podataka |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Stavka je uspješno izbrisana iz sustava |
| **Glavni uspješni scenarij:**   1. Korisnik iz pregleda svih stavki odabire onu koju želi obrisati 2. Sustav prikazuje prozor sigurnosnog upita 3. Korisnik potvrđuje brisanje 4. Sustav uspješno briše podatke 5. Korisnik se vraća na pregled knjige ulaznih računa |
| **Alternativni scenarij:**  3a. Korisnik odbacuje postupak brisanja stavke  3a1. Sustav ne provodi brisanje i korisnika vraća na pregled stavki  4a. Sustav zbog pogreške ne uspijeva pohraniti promjene  4a1. Sustav odustaje od promjena  4a2. Sustav dojavljuje korisniku poruku o pogreški |

|  |
| --- |
| **Naziv slučaja korištenja:** Stvaranje godišnjeg financijskog izvješća |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi stvoriti godišnje financijsko izvješće  Sustav – omogućava stvaranje izvješća i predaju u obliku datoteke |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Korisniku je vraćena datoteka izvješća |
| **Glavni uspješni scenarij:**   1. Korisnik odabire godinu izvješća i potvrđuje odabir 2. Sustav obavlja proračune i analizu korisnikovih knjigovodstvenih podataka 3. Sustav vraća podatke u obliku datoteke sa ispravno formatiranim tablicama |
| **Alternativni scenarij:**  Trenutno nema alternativnih scenarija |

|  |
| --- |
| **Naziv slučaja korištenja:** Stvaranje izvješća toka novca |
| **Primarni aktor:** Korisnik |
| **Dionici:**  Korisnik – želi stvoriti izvješće toka novca  Sustav – omogućava stvaranje izvješća i predaju u obliku datoteke |
| **Preduvjeti:**   1. Korisnik je prijavljen u sustavu |
| **Post-uvjeti:**   1. Korisniku je vraćena datoteka izvješća |
| **Glavni uspješni scenarij:**   1. Korisnik odabire godinu izvješća i potvrđuje odabir 2. Sustav obavlja proračune i analizu korisnikovih knjigovodstvenih podataka 3. Sustav vraća podatke u obliku datoteke sa ispravno formatiranim tablicama |
| **Alternativni scenarij:**  Trenutno nema alternativnih scenarija |

# Backend sustava

Kako je predviđena izgradnja sustava u obliku desktop i web aplikacije, implementacija web API-ja nije prijeko potrebna. *Backend* sustava će tako biti implementiran putem resursa koje obje aplikacije koriste. Potrebno je implementirati zajednički repozitorij podataka, zajednički izvor podataka, te zajednička funkcionalnost za generiranje izvješća.

Zajednički izvor podataka biti će u obliku SQLite baze podataka. Baza podataka će biti sadržana u jednoj SQLite datoteci.

Zajednički repozitorij podataka će stvarati vezu prema bazi podataka, te će vršiti O/R mapiranje podataka iz baze. O/R mapiranje će biti izvršeno NHibernate značajkom. Na taj način će zajednički repozitorij nuditi i CRUD metode, te još neke prilagođene metode dohvata. Repozitorij će također sadržavati osnovnu definiciju modela podataka koji će se koristiti u cijelom sustavu.

Kako oblici izvješća ne ovise o obliku aplikacije, funkcionalnost stvaranja izvješća se može izdvojiti i postaviti u pozadinu arhitekture sustava. Ovaj dio sustava će sadržavati svu logiku generiranja izvješća.

## Komponente aplikacije

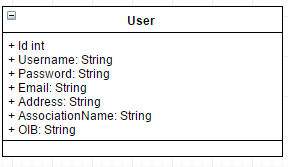
Objekti unutar aplikacije podijeljeni su u nekoliko vrsta prema zadaći koju obavljaju.

### Entitety

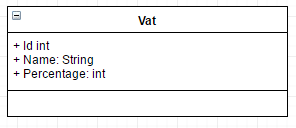
Objekti koji jedinstveno predstavljaju objekt aplikacije. Svaki entitet ima svoj jedinstveni broj koji ga razlikuje od ostalih objekata istog tipa.

Modeli unutar aplikacije su:

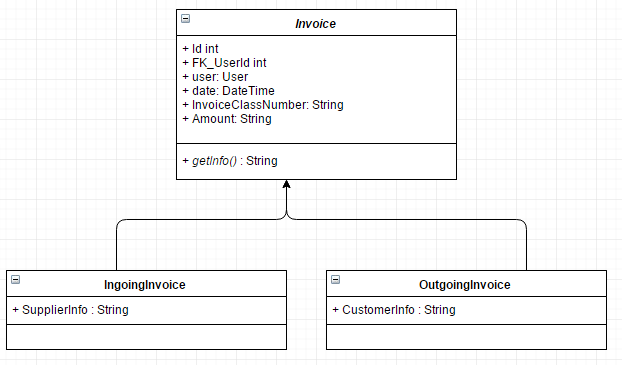
* User je model korisnika aplikacije. Sadrži sve bitne informacije o korisniku unutar aplikacije.



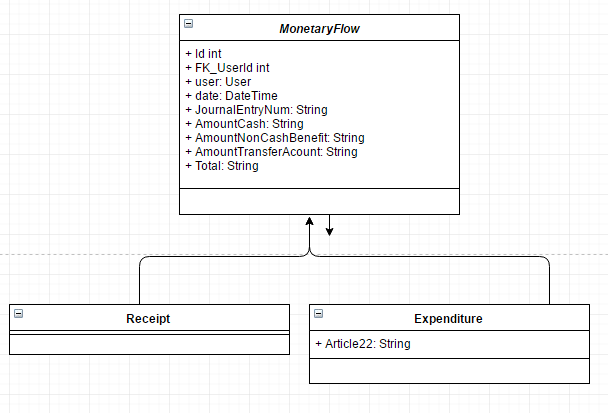
* Vat je model koji predstavlja PDV.



* Invoice – apstraktni model računa koji sadrži zajedničke elemente IngoingInvoice i OutgoingInvoice modela.

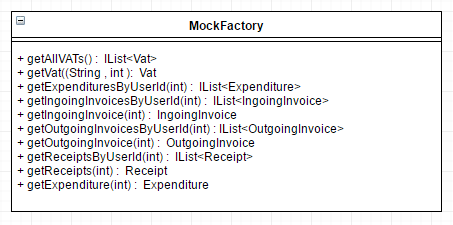


* MonetaryFlow je apstraktni model koji sadrži zajedničke elemente Receipt i Expenditure modela



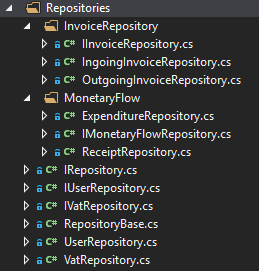
### Tvornice

MockFactory je klasa koja vraća mock objekte aplikacijskih modela.



### Repozitoriji

Repozitoriji su dijelovi aplikacije koji sadrže metode za dohvat objekata iz baze podataka koristeći nHibernate mogućnosti. Postoje repozitoriji za sve navedene entitete. Repozitorij implementira metode navedene u sučelju određenog repozitorija kako bi se omogućila laka zamjena konkretnog repozitorija nekom drugom vrstom implementacije. Postoji 6 repozitorija i 4 sučelja. Repozitoriji su: IngoingInvoiceRepository, OutgointInvoiceRepository, ExpenditureRepository, ReceiptRepository, UserRepository, VatRepository,a sučelja su: IUserRepository, IVatRepository, IMonetaryFlowRepository i IinvoiceRepository



Navest ćemo primjer implementacije korisničkog repozitorija i sučelja za korisnički repozitorij.

IUserInterface je sučelje koje sadrži metode koje implementira UserRepository repozitorij.

public interface IUserRepository

{

void Create(User user);

User GetById(int id);

User GetUserByCredentials(UserCredentials userCredentials);

void UpdateUser(User user);

void DeleteUser(User user);

}

public class UserRepository : IuserRepository

ISessionFactory objekt unutar repozitorija je zajednički objekt za sve pozivane metode. Kreira se na početku kreiranja repozitorija i ima istu vrijednost tijekom cijelog korištenja objekta repozitorija. Proizvoljnu implementaciju IsessionFactory objekta moguće je predati objektu preko konstruktora. Ukoliko je konstruktor prazan, koristi se predefinirani ISessionFactory objekt.

private ISessionFactory sessionFactory;

public UserRepository(ISessionFactory sessionFactory)

{

this.sessionFactory = sessionFactory;

}

public UserRepository()

{

this.sessionFactory = SessionManager.SessionFactory;

}

Svaka metoda repozitorija obavlja zadanu radnju koristeći transakcije i sesije.

public void Create(User user)

{

using (var session = sessionFactory.OpenSession())

{

using (ITransaction transaction = session.BeginTransaction())

{

session.SaveOrUpdate(user);

transaction.Commit();

}

}

}

## Perzistencija kroz O/R mapiranje

Perzistencija se provodi koristeći O/R mapiranje s Fluent Nhibernateom u SQLite bazu podataka.

Fluent Nhibernate za razliku od običnog Nhibernatea ne provodi mapiranje putem XML-a već unutar C# koda. Predefiniranim setom naredbi objekti baze se prevode u objekte koji se mogu koristiti unutar aplikacije.

UserMap je model koji mapira User klasu.

public class UserMap : ClassMap<User>

{

public UserMap()

{

Table("User");

Id(x => x.Id).GeneratedBy.Native().Unique();

Map(x => x.Username).Not.Nullable();

Map(x => x.Password).Not.Nullable();

Map(x => x.Email).Nullable();

Map(x => x.Address).Nullable();

Map(x => x.AssociationName).Nullable();

Map(x => x.OIB).Nullable();

}

}

InvoiceMap je model koji mapira Invoice klasu.

public class InvoiceMap : ClassMap<Invoice>

{

public InvoiceMap()

{

Id(x => x.Id).GeneratedBy.Increment();

Map(x => x.Date);

Map(x => x.Amount);

Map(x => x.InvoiceClassNumber);

References(c => c.User).Column("FK\_UserId").Not.LazyLoad();

Map(x => x.FK\_UserId).Formula("[FK\_UserId]");

UseUnionSubclassForInheritanceMapping();

}

}

IngoingInvoiceMap i OutgointInvoiceMap su modeli koji mapiraju IngoingInvoice i OutgoingInvoice koristeći zajedničke elemente iz InvoiceMap modela.

public class IngoingInvoiceMap : SubclassMap<IngoingInvoice>

{

public IngoingInvoiceMap()

{

Map(x => x.SupplierInfo);

}

}

VatMap je klasa koja mapira Vat model.

public class VatMap : ClassMap<Vat>

{

public VatMap()

{

Id(x => x.Id).GeneratedBy.Native().Unique();

Map(x => x.Name).Not.Nullable();

Map(x => x.Percentage).Not.Nullable();

Table("VAT");

}

}

MonetaryFlowMap je klasa koja mapira abstraktnu klasu MonetaryFlow. Sadrži zajedničke elemente od Receipt i Expenditure klase.

public class MonetaryFlowMap : ClassMap<MonetaryFlow>

{

public MonetaryFlowMap()

{

Id(x => x.Id).GeneratedBy.Increment();

Map(x => x.Date);

Map(x => x.AmountCash);

Map(x => x.AmountNonCashBenefit);

Map(x => x.AmountTransferAccount);

References(c => c.Vat).Column("FK\_VAT").Not.LazyLoad();

References(c => c.User).Column("FK\_UserId").Not.LazyLoad();

Map(x => x.FK\_UserId).Formula("[FK\_UserId]");

Map(x => x.FK\_VAT).Formula("[FK\_VAT]");

Map(x => x.JournalEntryNum);

Map(x => x.Total);

UseUnionSubclassForInheritanceMapping();

}

}

ReceiptMap je klasa koja mapira Receipt model.

public class ReceiptMap : SubclassMap<Receipt>

{

}

ExpenditureMap je klasa koja mapira Expenditure model.

public class ExpenditureMap : SubclassMap<Expenditure>

{

public ExpenditureMap()

{

Map(x => x.Article22);

}

}

# Mockup zasloni

## Desktop aplikacija

### Glavni zaslon



Glavni zaslon desktop aplikacije će sadržavati izbornike padajućih lista, koje će omogućiti korisniku pristup pojedinim knjigovodstvenim knjigama, generiranju izvješća i odjavi sa sustava. Zasloni pozvanih funkcionalnosti će se korisniku pojavljivati uz glavni zaslon.

### Zaslon generiranja izvješća



Zaslon generiranja izvješća će imati jednu padajuću listu, gdje će korisnik moći odabrati godinu izvješća, te gumb za pokretanje stvaranja izvješća. Na akciju pokretanja izvješća predviđeno je korisniku prikazati dijaloški okvir za spremanje datoteke.

### Zasloni pregleda i uređivanja podataka





Na zaslonu pregleda stavki knjigovodstvene knjige biti će prikazana tablica svih stavki. Korisniku će biti ponuđen gumb za stvaranje nove stavke, koji će pozvati dijaloški okvir sa formom za unos podataka. Ukoliko korisnik odabere neki red tablice pregleda, biti će mu omogućeni gumbi za brisanje i uređivanje odabrane stavke.

Dijaloški okviri za stvaranje nove i uređivanje stavke će na pripadajućim mjestima imati padajuće kalendare za izbor datuma i padajuće liste za definirana svojstva. Biti će omogućeno spremanje izmjena putem gumba, ali i odustajanje od izmjena. Korisniku će također biti dostupne validacijske poruke.

## Web aplikacija

### Glavni zaslon





Glavni zaslon će sadržavati padajuće izbornike preko kojih će korisnik moći pristupiti svim funkcionalnostima sustava: uređivanje knjigovodstvenih podataka, stvaranje izvješća i odjava sa sustava.

### Zaslon generiranja izvješća



Zaslon generiranja izvješća će omogućiti korisniku odabir godine izvješća putem padajuće liste, te gumb pomoću kojeg će pokretati postupak. Korisnik će u slučaju web aplikacije direktno zaprimiti datoteku (*download*).

### Zaslon pregleda podataka



Na zaslonu pregleda podataka biti će prikazane sve stavke odabrane knjigovodstvene knjige u obliku tablice. Korisniku će biti ponuđen gumb za stvaranje nove stavke (koji će ga dovesti do zaslona forme za unos podataka). Uz svaku stavku stajati će njoj pripadajući gumbi za uređivanje i brisanje. Gumb za uređivanje će korisnika dovesti do zaslona forme za unos podataka – u ovom slučaju će forma biti popunjena postojećim podacima.

### Zaslon za unos i uređivanje podataka



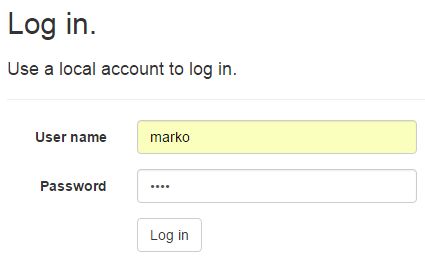
Na zaslonu za unos i uređivanje podataka će biti prikazana polja pojedinih atributa stavke. Netrivijalna polja za unos će biti potpomognuta padajućim kalendarima i padajućim izbornicima. Postojati će gumb za odustajanje od promjena i gumb za spremanje promjena u sustav. Uz polja za unos po potrebi će se pojavljivati validacijske poruke.

# Opis implementirane web aplikacije

Web aplikacija ovog projekta je implementirana uz pomoć ASP.NET MVC 5 tehnologije, uz .NET Framework 4.5.2. Aplikaciji se može pristupiti putem bilo kojeg modernog web pretraživača.

U nastavku će biti opisani zasloni koji najbolje opisuju funkcionalnosti sveukupne aplikacije.

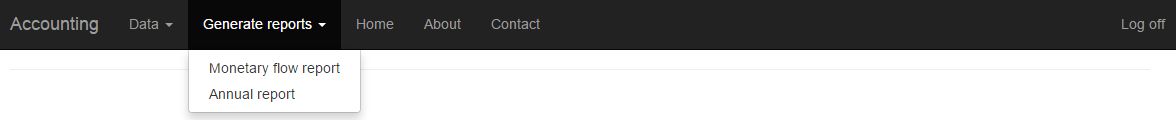
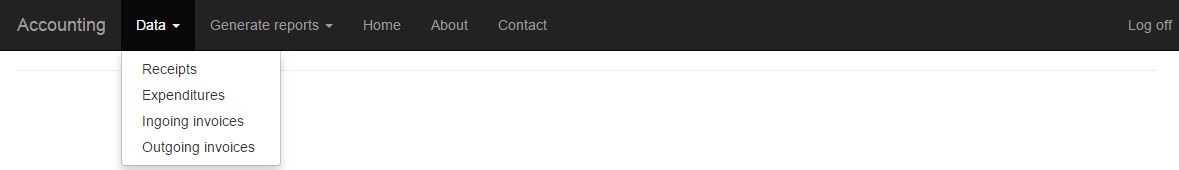
## Zaslon prijave na sustav



Zaslon prijave na sustav omogućuje korisniku unos svojeg korisničkog imena i lozinke, te prijavu na sustav putem gumba *Log in* ukoliko su vjerodajnice ispravne.

## Izbornik

Nakon uspješne prijave korisniku se pojavljuje izbornik preko kojega može pristupiti svojim knjigovodstvenim podacima.

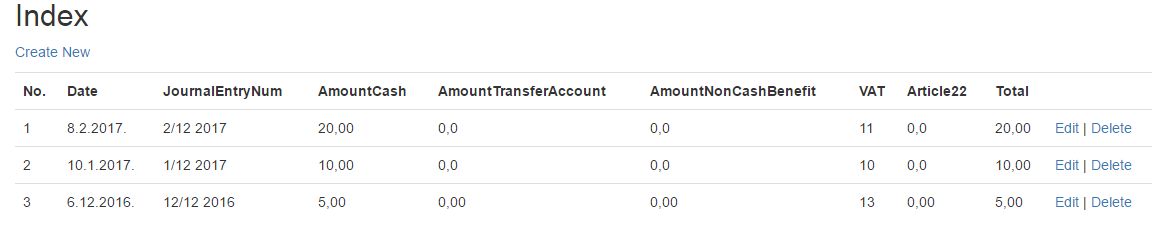


Preko *Data* padajućeg izbornika korisnik može pristupiti svojim knjigovodstvenim knjigama. Preko padajućeg izbornika *Generate reports* korisnik može pristupiti dvjema vrstama izvješća (izvješće toka novca i godišnje financijsko izvješće).

Također su implementirani i *Home*, *About* i *Contact* gumbi, zbog mogućih predočenja podataka o sustavu i onima koji ga održavaju.

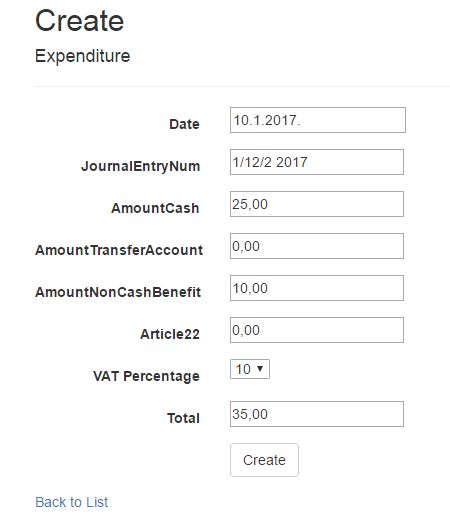
Sa krajnje desne strane (nije vidljivo u slikama) stoji *Log off* gumb koji omogućuje odjavu korisnika sa sustava.

## Pregled knjigovodstvene knjige



Na slici je prikazan pregled knjige izdataka. Zdesna svake stavke stoje dva gumba *Edit* i *Delete* koji se odnose na tu stavku. *Edit* omogućuje uređivanje dane stavke, dok *Delete* briše stavku. Iznad tablice prikaza stavki nalazi se gumb *Create New* koji omogućuje korisniku stvaranje novih stavki knjige.

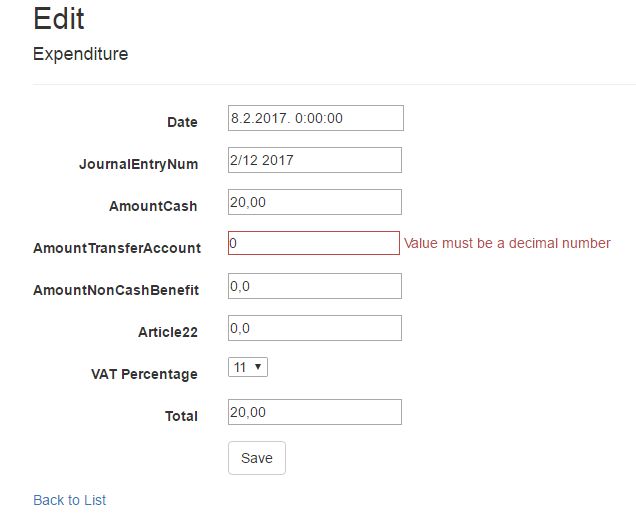
## Dodavanje nove stavke



Odabirom stvaranja nove stavke korisniku se predstavlja prazna forma za unos podataka o novoj stavki knjige (u slučaju slike, podaci su već ispravno popunjeni). Za polje *Date* korisniku je omogućen izbornik datuma (*Datepicker*), te je za postotak PDV-a (*VAT Percentage*) ponuđena padajuća lista svih propisanih postotaka. Ukoliko neko polje nije ispravno popunjeno, korisniku se crvenom bojom prikazuju validacijske poruke.

Klikom na gumb *Save*, podaci iz forme se pohranjuju u sustav (uz uvjet da je forma uspješno validirana). Ukoliko podaci nisu uspješno validirani tokom pritiska na gumb *Save*, korisnik ostaje na prikazanoj formi uz prikaz validacijskih poruka. Korisnik može odustati od unosa klikom na *Back to List*, pritom se validacija ignorira, ali se ni uneseni podaci ne pohranjuju.

## Uređivanje postojeće stavke



Ukoliko je korisnik odabrao uređivanje neke stavke, predstavlja mu se forma sa već popunjenim trenutnim podacima. Kao i u slučaju forme stvaranja, korisnik se može služiti padajućim listama i izbornikom datuma na prikladnim mjestima. Uneseni podaci prolaze validaciju, te se polja neispravno unesenih podataka obrube crvenom bojom uz prikaz validacijske poruke.

Gumb *Save* pokreće pohranu podataka, uz uvjet da je validacija podatak bila uspješna. U suprotnom korisnik ostaje na formi i prikazuju mu se validacijske poruke. Gumb *Back to List* služi za odustajanje od izmjena unesenih u formu, te njegova funkcionalnost ne ovisi o validaciji podataka.

## Izvješća



Zasloni za zahtjeve izvješća sadrže padajući izbornik sa svim godinama za koje je moguće generirati dano izvješće. Pritiskom na gumb *Generate report* pokreće se postupak stvaranja izvješća, te se korisniku vraća datoteka izvješća (za sada samo u HTML obliku).

## Opis korištene MVC arhitekture

Ova aplikacija je građena prateći MVC (*model-view-controller*) uzorak izgradnje web sustava. Primijenjeno je proširenje nad standardnom MVC arhitekturom dodatkom *BindingModels* sloja.

*BindingModels* (modeli vezivanja) su služili za prijenos podataka u *view*, te lakše stvaranje padajućih lista. Unutar klase *binding model-a* moguće je pronaći svojstvo u obliku klase modela koji se mora prikazati, te svojstva lista podataka koji se prikazuju na korisničkom sučelju kao padajuće liste.

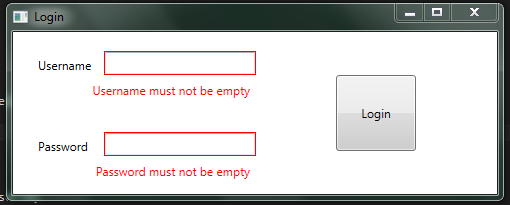
Umjesto standardnog pristupa u ASP.NET MVC-u (gdje se model nalazi unutar samog projekta), model podataka je uzet iz projekta *DataRepository*, gdje se nalaze potrebni Repozitoriji i klase koje opisuju model baze podataka.

# Opis implementirane Desktop aplikacije

Desktop aplikacija ovog projekta je implementirana korištenjem tehnologije Windows Presentation Forms uz korišten oblikovni obrazac Model View ViewModel (MVVM), uz .NET framework 4.5.2.

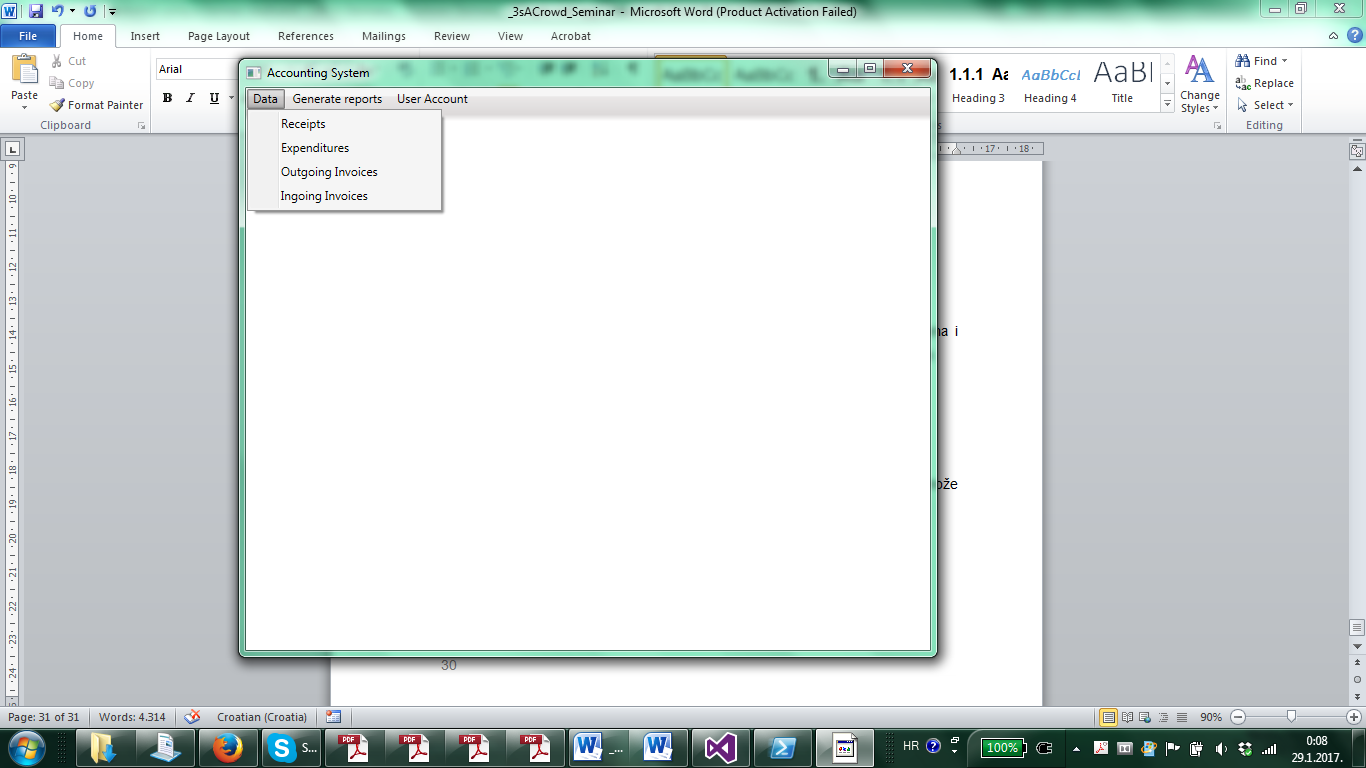
U nastavku će biti opisani zasloni koji najbolje opisuju funkcionalnosti sveukupne aplikacije.

## Zaslon prijave na sustav

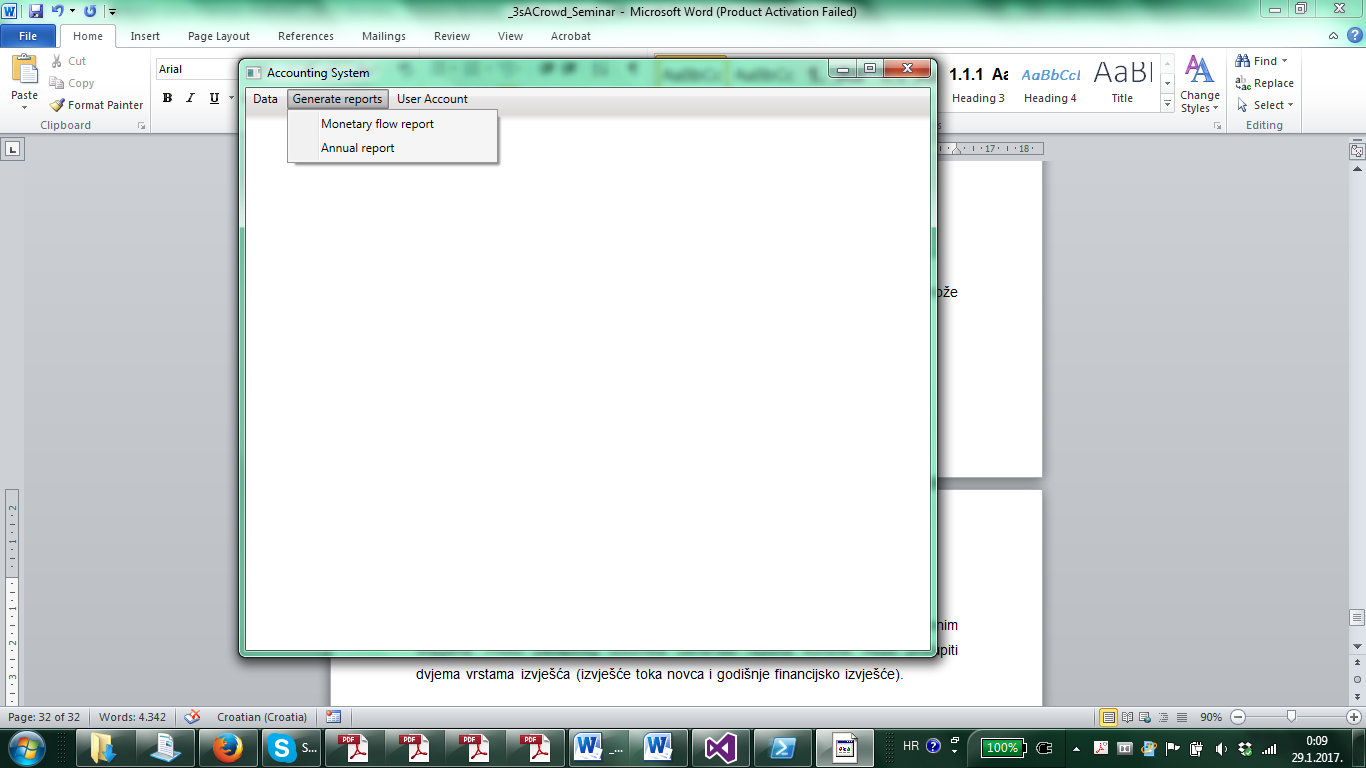


Zaslon prijave na sustav omogućuje korisniku unos svojeg korisničkog imena i lozinke, te prijavu na sustav putem gumba *Log in* ukoliko su vjerodajnice ispravne.

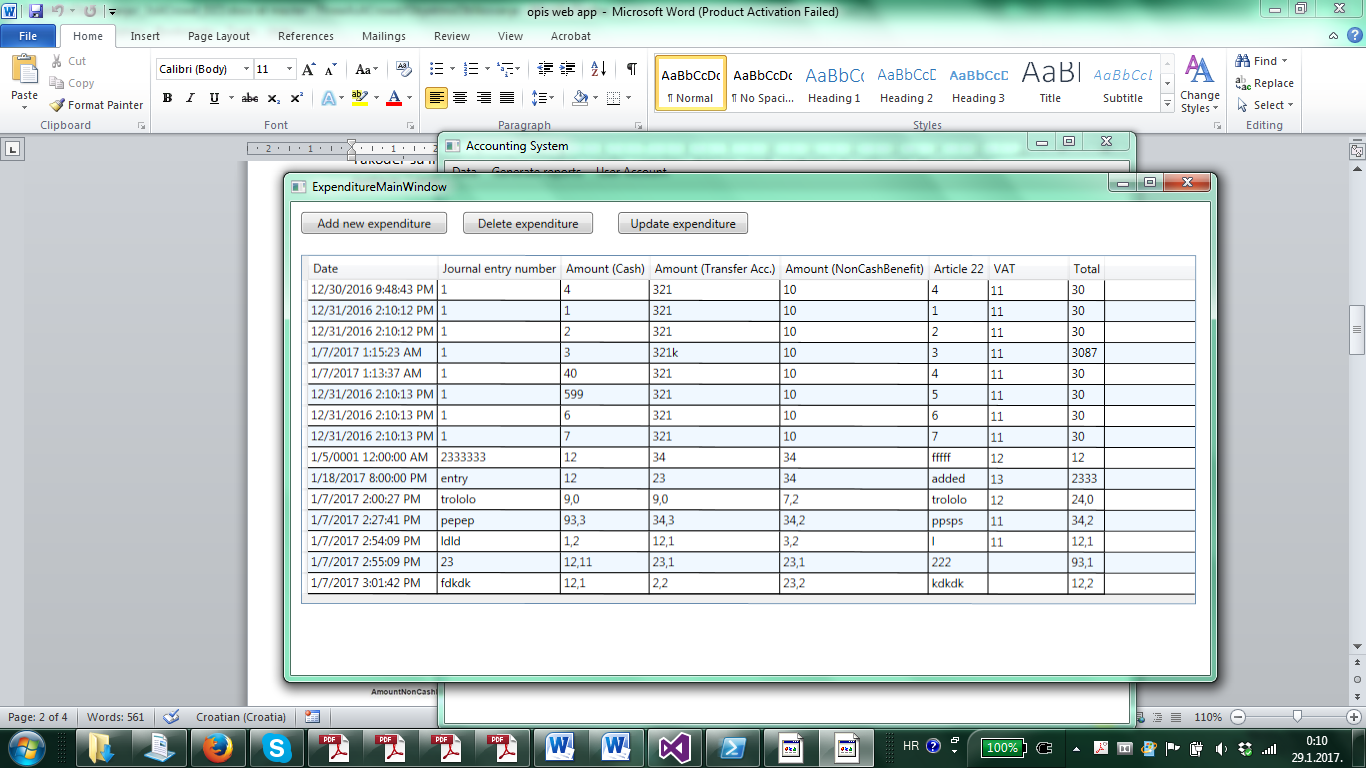
## Izbornik



Nakon uspješne prijave korisniku se pojavljuje izbornik preko kojega može pristupiti svojim knjigovodstvenim podacima.

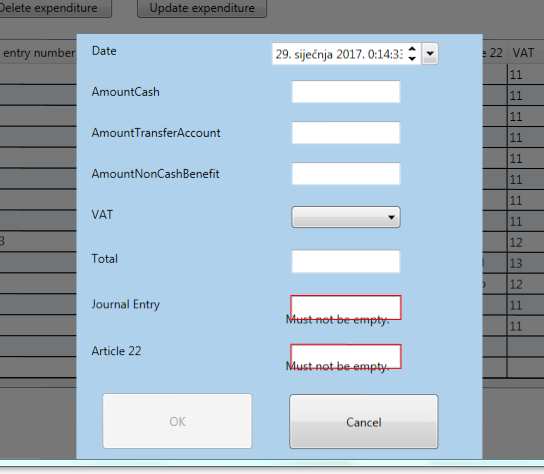


Preko *Data* padajućeg izbornika korisnik može pristupiti svojim knjigovodstvenim knjigama. Preko padajućeg izbornika *Generate reports* korisnik može pristupiti dvjema vrstama izvješća (izvješće toka novca i godišnje financijsko izvješće).



Na slici je prikazan pregled knjige izdataka. Gumb *Add new expenditure* otvara novu formu za unos nove stavke. Selekcijom nekog unosa, moguće je njegovo ažuriranje klikom na gumb *Update expenditure*, također otvaranjem posebne forme. Također je moguće brisanje odabrane stavke klikom na gumb *Delete expenditure*.

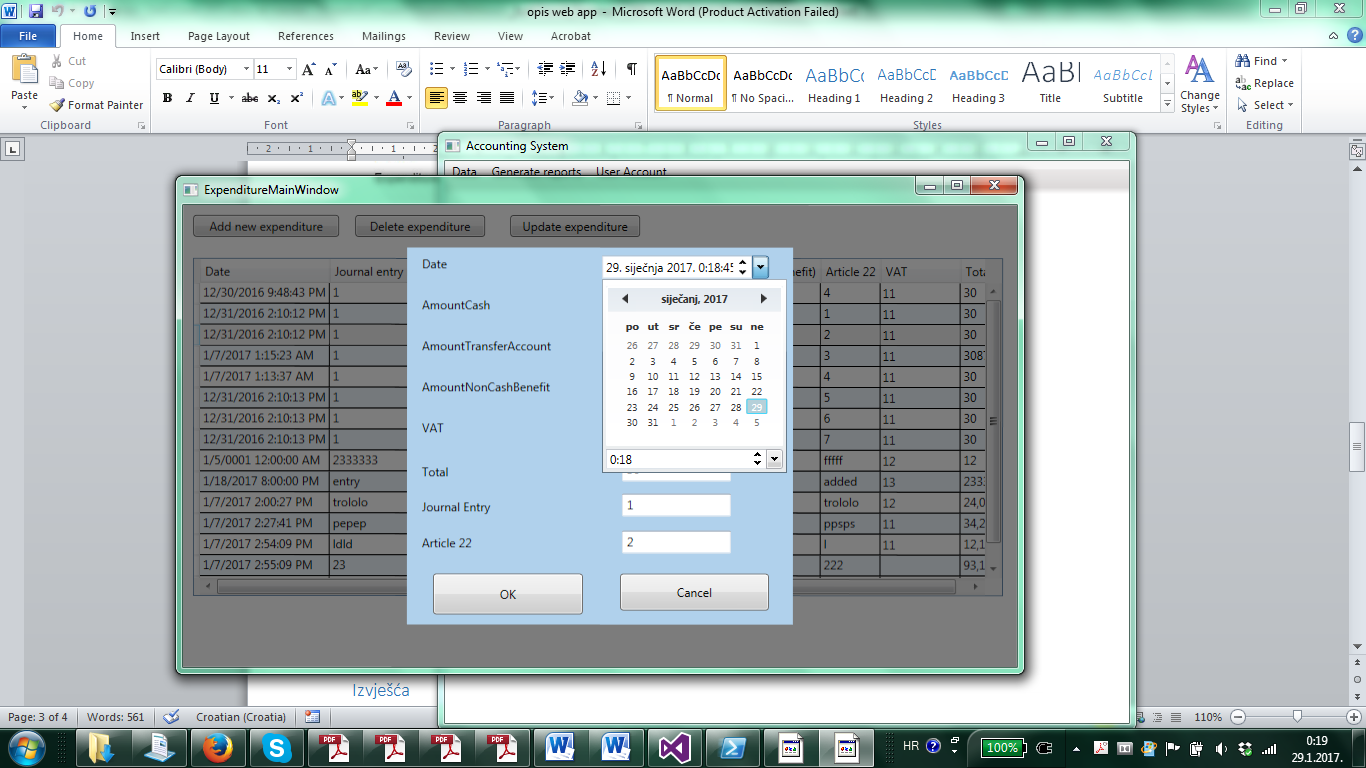
## Dodavanje nove stavke



Odabirom stvaranja nove stavke korisniku se predstavlja prazna forma (tipa *User Control*) za unos podataka o novoj stavki knjige. Za polje *Date* korisniku je omogućen izbornik datuma (*Datepicker*), te je za postotak PDV-a (*VAT Percentage*) ponuđena padajuća lista svih propisanih kategorija poreza na dodanu vrijednost. Ukoliko neko polje nije ispravno popunjeno ili je prazno, a ne smije biti, korisniku se crvenom bojom prikazuju validacijske poruke.

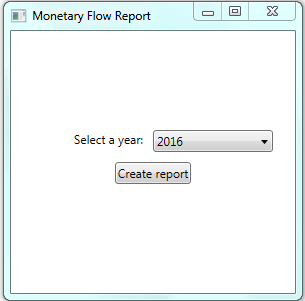
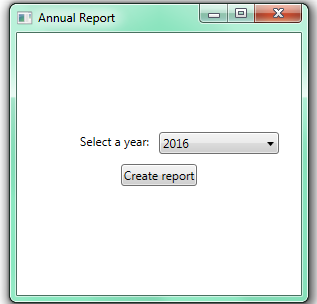
Klikom na gumb *Ok*, podaci iz forme se pohranjuju (uz uvjet da je forma uspješno validirana). Ukoliko podaci nisu uspješno validirani tokom pritiska na gumb *Ok*, korisnik ostaje na prikazanoj formi uz prikaz validacijskih poruka. Korisnik može odustati od unosa klikom na *Cancel*, pritom se validacija ignorira, ali se ni uneseni podaci ne pohranjuju.

## Ažuriranje podataka u postojećoj stavci

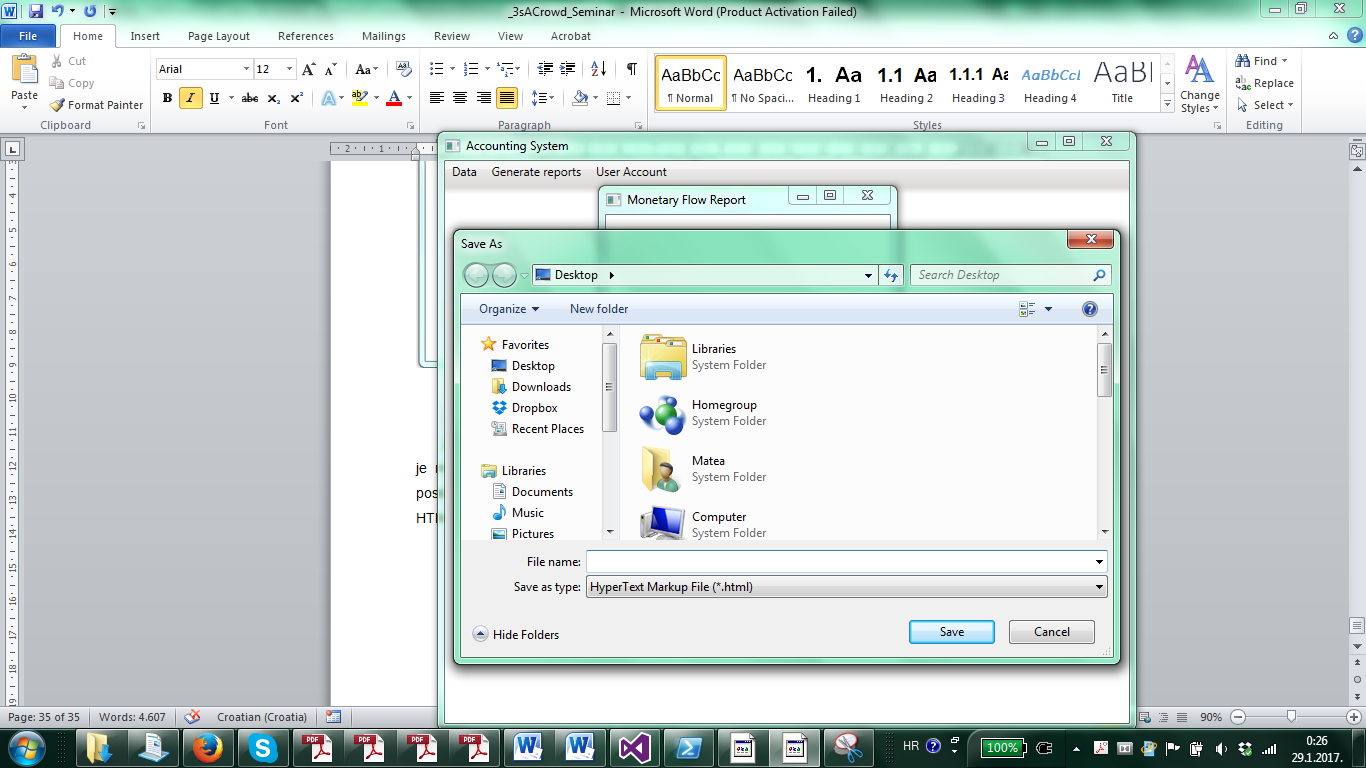


Ukoliko je korisnik odabrao uređivanje neke stavke, predstavlja mu se forma sa već popunjenim trenutnim podacima. Kao i u slučaju forme stvaranja, korisnik se može služiti padajućim listama i izbornikom datuma na prikladnim mjestima. Gumb *Ok* pokreće pohranu podataka, a Gumb *Cancel* služi za odustajanje od izmjena unesenih u formu.

## Izvješća

Zasloni za zahtjeve izvješća sadrže padajući izbornik sa svim godinama za koje je moguće generirati dano izvješće. Pritiskom na gumb *Create report* pokreće se postupak stvaranja izvješća, te se korisniku vraća datoteka izvješća u HTML obliku.



## Opis iskorištene MVVM arhitekture

Umjesto standardnog pristupa u MVVM arhitekturi (gdje se model nalazi unutar samog projekta), model podataka je uzet iz projekta *DataRepository*, gdje se nalaze potrebni Repozitoriji i klase koje opisuju model baze podataka.

Prilikom izgradnje Desktop aplikacije, za implementaciju MVVM oblikovnog obrasca, iskoristila se Microsoft.Practices.Prism programska biblioteka, za lakšu implementaciju ViewModela kod User Controla za dodavanje nove stavke te ažuriranje neke označene stavke u knjizi. Za brisanje neke označene stavke, koristila se vlastita implementacija nad sučeljem ICommand.