

**PROCES RAZVOJA GENERIČKE POSLOVNE APLIKACIJE
DEVELOPMENT OF GENERIC BUSINESS APPLICATION**Ilija Grbić, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U današnje vreme sve više jača tendencija digitalizacije podataka iz različitih domena kako bi se olakšala njihova pretraga i brz pristup. Upotreba informacionih sistema postala je opšteprihvaćena u oblastima obrazovanja, medicine i privrede. Državna uprava i pravni sistem u Republici Srbiji postepeno primenjuju informacione sisteme u svakodnevnom poslovanju što se dalje odražava i na advokate. Advokatske kancelarije trenutno prvenstveno rade sa fizičkim ili eventualno skeniranim dokumentima, dok je primena namenskih softverskih alata slabo prisutna. Cilj ovog rada jeste prikaz procesa razvoja delom generičke aplikacije za evidentiranje i upravljanje pre svega advokatskih predmeta, ali i ostalih podataka neophodnih u radu advokatske kancelarije.

Gljučne reči: *Objektno orijentisano programiranje, veb aplikacije, Objektno orijentisani šabloni, Softver za pravnike*

Abstract – In modern times it is more and more common for data from varying domains to be stored in digital format for faster search and access. Usage of information systems is becoming defacto standard in education, medicine and economy. State Administration and legal system in the Republic of Serbia gradually apply information systems in everyday use, which in turn reflects on lawyers. Law firms currently mainly use physical documents and sometimes scanned documents, while application of domain specific software is quite rare. The goal of this paper is to summarize the development of partly generic application for case management and other necessary lawyer data.

Keywords: *Object oriented programming, Web applications, Object oriented patterns, Legal software*

1. UVOD

Upravljanje podacima i njihova evidencija predstavlja jedan od glavnih izazova sa kojima se advokati sadašnjice susreću. Svake godine se u sudove u Republici Srbiji predaju stotine hiljada predmeta, konkretno u prvih pet najvećih sudova primljeno je preko 400.000 [1]. Iako nijedan advokat ne brine ni blizu o tolikoj količini predmeta, jasno je da je pri radu sa tolikim podacima korisno primeniti tehnike iz oblasti velikih podataka (eng. *big data*) i nauka o podacima (eng. *data science*).

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Aleksandar Kupusinac, vanredni prof.

Ove tehnike se primenjuju kada postoje prikupljeni podaci u velikoj količini, što dodatno daje na značaju sistemima za evidenciju podataka.

Prilikom postavljanja zahteva za razvoj aplikacija, pored komunikacije sa potencijalnim korisnicima, neophodno je analizirati postojeća rešenja u datoj oblasti. U oblasti pravnih sistema, uže namenjenih advokatima postoji mnoštvo stranih i domaćih rešenja. Najkorištenija rešenja na stranom tržištu su *PracticePanther* i *Clio* koja nude mnoštvo funkcionalnosti poput personalizovanog rada, automatske naplate, planiranja događaja, evidencije predmeta itd. Na domaćem tržištu korišćenje ovakvog namenskog softvera nije u velikoj meri zastupljeno. Advokatske kuće i dalje koriste softver drugih namena, poput *Paragraf Lex* koji je pre svega namenjen za pretragu baze podataka zakona.

2. RAZMATRANI ZAHTEVI

Razmatrane zahteve ćemo podeliti u dve grupe: esencijalni i opcioni. Podela je kreirana pre svega da bi se olakšala implementacija i što pre dobio prototip aplikacije koji podržava samo esencijalne mogućnosti.

Esencijalni zahtevi su:

1. Generička evidencija advokatskih predmeta
2. Registracija i prijava korisnika
3. Upravljanje korisničkim profilom
4. Evidencija kontakata
5. Otpremanje dokumenata
6. Interaktivno planiranje aktivnosti.

Opcioni zahtevi su:

1. Generisanje izveštaja za naplatu
2. Evidencija sudova
3. Brza pretraga dokumenata
4. Konverzija skeniranih dokumenata u tekst.

Generička evidencija predmeta je potrebna jer tipovi predmeta koji mogu da se pamte u sistemu nisu uvek isti. Metapodaci za različite predmete mogu biti značajno drugačiji tako da je jedan od glavnih problema kreiranje sistema koji može pamtititi različite vrste predmeta kao i njihove metapodatke. Cilj je da broj tipova predmeta i njihovi metapodaci ne budu predefinisani, već da mogu generički da se podese za odgovarajući region ili konkretne potrebe nekog korisnika.

Kako bi se obezbedila personalizovano radno iskustvo na aplikaciji, registracija i prijava korisnika je neophodna. Takođe ovo potencijalno može biti korisno i u slučaju kada je neophodno ograničiti pristup podacima u sistemu određenim korisnicima.

Potrebno je omogućiti generisanje izveštaja za naplatu na osnovu podataka u sistemu. Ovi izveštaji bi trebalo da se koriste dalje za naplatu usluga u skladu sa definisanim cenovnicima. Bitan detalj za ovaj slučaj korišćenja jeste omogućavanje personalizovanja šablona za kreiranje izveštaja u skladu sa zahtevima pojedinačnog korisnika.

3. KORIŠTENE TEHNOLOGIJE

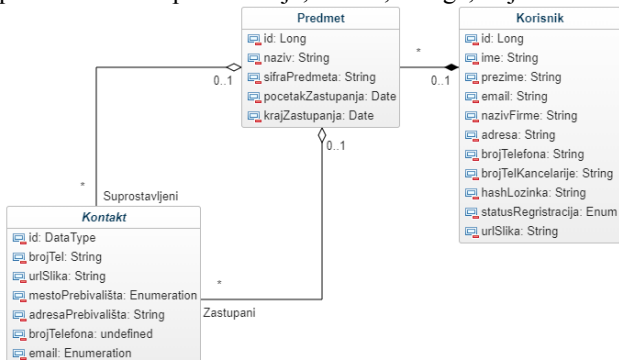
Sam softverski proizvod je realizovan kao *web* aplikacija. Svaka *web* aplikacija se sastoji od *backend*-a i *frontend*-a.

Backend je realizovan u *Java* programskom jeziku uz upotrebnu *Spring framework*. Za čuvanje podataka se koristi relaciona baza podataka. *Backend* komunicira sa ostatkom sistema putem HTTP zahteva kreiranih po REST arhitektonskom šablonu. Za generisanje izveštaja se koristi *JasperReports* biblioteka koja omogućava generisanje izveštaja na osnovu predefinisanih XML šablona [2]. Ovi šabloni su značajni jer omogućavaju personalizovano kreiranje šablona za na osnovu potreba pojedinačne advokatske kuće. *Elasticsearch* se koristi za indeksiranje dokumenata koje omogućava brzu pretragu otpremljenih tekstualnih dokumenata ne samo po nazivu već i po njihovom sadržaju [3].

Frontend je realizovan kao klijentska jednostranična *Angular* aplikacija (eng. *single-page application*). *Angular* predstavlja jedan od najpopularnijih *framework*-a, pored *ReactJS* i *VueJS*, za kreiranje *web* frontenda [4].

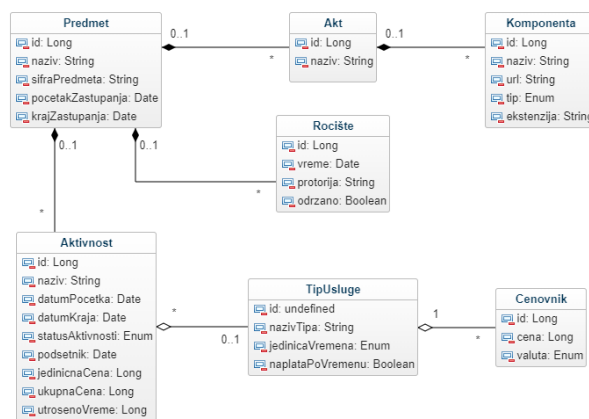
4. GENERIČKI PREDMET

Advokatski predmet je modelovan tako da svaki korisnik sistema ima listu predmeta na kojima radi. Sam predmet ima dalje sve relevantne podatke poput zastupane i suprostavljene strane, sudija angažovanih na slučaju, statusa predmeta, akata, ročišta i aktivnosti. Na slici 1. može se videti deo ovih metapodataka. Klasa *Kontakt* reprezentuje podatke o osobi od značaja u advokatskom sistemu i može imati proizvoljan tip, dok su neki od predefinisanih tipova: sudija, veštak, kolega, klijent i sl.



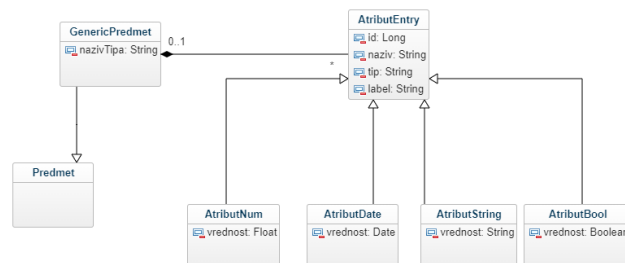
Slika 1. Deo klas modela predmeta

Aktivnosti predstavljaju jednu od najbitnijih delova sistema jer omogućavaju evidenciju svih bitnih podataka vezanih za naplatu advokatskih usluga. Aktivnost može imati proizvoljan tip kako bi se omogućila fleksibilnost korisnicima prilikom njihovog definisanja. Ovo je modelovano klasom *TipUsluge* Omogućeno je i definisanje cena aktivnosti u zavisnosti od specifičnosti konkretnog predmeta, koje omogućava klasa *Cenovnik*.



Slika 2. Klas model veze predmeta, akata, ročišta i aktivnosti

Predmet pored pored prethodno navedenih metapodataka može imati dodatne metapodatke koji mogu biti različiti u zavisnosti od tipa predmeta [5]. Ovaj problem bi se mogao rešiti pomoću nasleđivanja gde bi se za svaki tip predmeta definisali dodatna neophodna polja. Međutim, ovakav pristup ima nekoliko problema, najočiglednije za svaki novi tip predmeta koji treba podržati morao bi se menjati izvorni kod i implementirati odgovarajuću klasu. Pored toga što sudski pravilnik može da se promeni, ukoliko želimo da podržimo upotrebljivost aplikacije i na stranom tržištu tj. u drugim pravnim sistemima, nepraktično je implementirati sve moguće predmete. Ovaj problem može biti rešen implementacijom generičke klase koja bi opisala bilo koji predmet, uz odgovarajuća ograničenja.



Slika 3. Klas model generičkog predmeta

Generički predmet pored nasleđenih polja iz klase *Predmet* ima listu dodatnih atributa. Svaki od ovih atributa ima naziv atributa, jedinstven na nivou jednog predmeta, labelu za prikaz na korisničkom interfejsu, tip podatka koji se čuva, kao i polje koje govori da li da se atribut prikazuje u skraćenom prikazu na korisničkom interfejsu.

Do sad je definisan samo način da se generički predmet sačuva, ali ne i način da se kreira. Prvi problem koji se nameće je definisanje metapodataka za sve predmete koje treba čuvati unutar sistema. Ovo je moguće rešiti ubacivanjem tih podataka u bazu podataka, međutim, prosečni korisnik, pa čak i administrator sistema, ne bi trebao da komunicira sa sistemom na ovom nivou. Kako bi se ovaj problem rešio definisanje podataka o podržanim predmetima je realizovano pomoću konfiguracionog fajla. Za format konfiguracionog fajla je odabran JSON, pre svega zbog lake ljudske čitljivosti fajla.

```

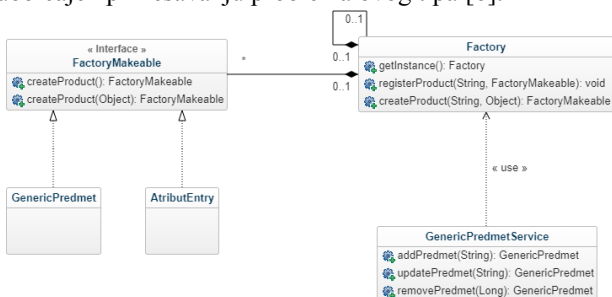
{
  "predmeti": [
    {
      "tip": "Privredni",
      "atributi": [
        {
          "naziv": "kazna",
          "label": "Kazna",
          "prikazi": false,
          "atr_tip": "String"
        },
        {
          "naziv": "vrstaOdluke",
          "label": "Vrsta odluke",
          "prikazi": true,
          "atr_tip": "Enum",
          "enum_values": [
            "USVOJEN",
            "ODBIJEN"
          ]
        }
      ]
    }
  ],
}

```

Slika 4. Primer konfiguracionog fajla za jednostavan tip predmeta

Konfiguracioni fajl se sadrži od liste predmeta koje korisnik želi da podrži gde svaki predmet ima naziv tipa i atribute koji su za njega neophodni. Polje *atr_tip* je obavezno i definiše tip atributa. Podržani tipovi su podržani su numerički (poput *Float* ili *Long*), datum (*Date*), proizvoljan tekst (*String*), logički tip (*Boolean*) i nabrojivi tip (*Enum*). Jedino za nabrojivi tip su potrebni dodatni metapodaci tj. lista svih opcija (*enum_values* lista). Sistem učitava konfiguracioni fajl i na osnovu njega zna koji tipovi predmeta su podržani kao i njihovu strukturu.

Kreiranje objekata klase *GenericPredmet* zahteva i generisanje objekata klase *AtributEntry* (slika 3.) u realnom vremenu. Kako bi se kreiranje što elegantnije realizovalo, potrebna je primena *Factory* šablona koji je uobičajen pri rešavanju problema ovog tipa [6].

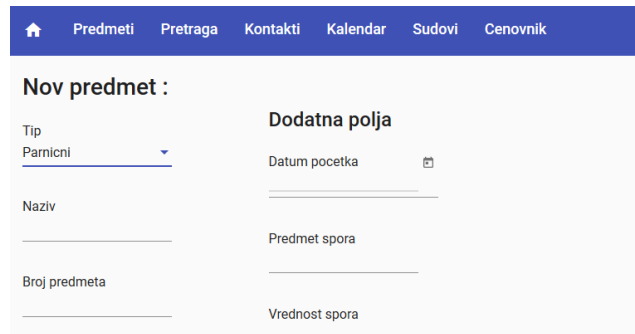


Slika 5. Klas model fabrika šablona za kreiranje generičkih predmeta

5. KORISNIČKI INTERFEJS

Bitan zahtev prilikom kreiranja korisničkog interfejsa je mogućnost da on funkcioniše na različitim uređajima. *Angular framework* podržava korišćenje *AngularMaterial* komponenti koje pružaju dobru podršku za prikaz kako na mobilnim uređajima poput telefona i tableta tako i na personalnim računarima.

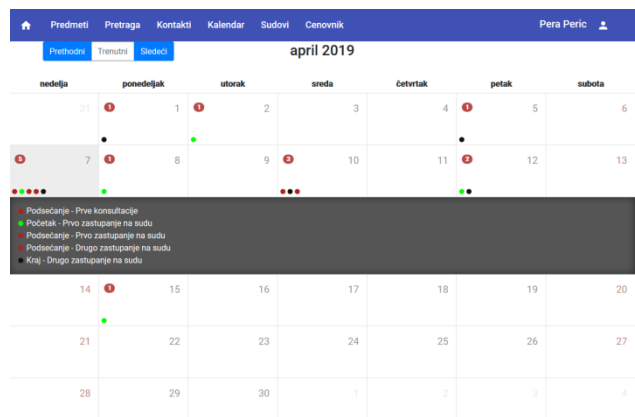
Prilikom unosa predmeta pored podrazumevanih polja za svaki predmet potrebno je uneti i dodatna polja definisana u tipu generičkog predmeta.



Slika 6. Prikaz dela korisničkog interfejsa za unos predmeta

Na slici 6. se može videti primer izgleda ovakvog korisničkog interfejsa. Nakon što korisnik izabere tip predmeta, sa desne strane se prikazu izgenerisanja polja za unos specifičnih atributa.

Jedna od najbitnijih funkcionalnosti je prikaz aktivnosti u nekom vremenskom periodu na lak i intuitivan način. Ovo je realizovano prikazivanjem aktivnosti u formi kalendara. Svaka aktivnost ima tri bitne vremenske odrednice: početak, kraj i vreme podsećanja. Svako od ovih vremena je prikazano u kalendaru odgovarajućom bojom: početak zelenom, kraj crnom i podsećanje crvenom (slika 7.). Korisnik svako od vremena može *drag-n-drop* interakcijom da promeni. Aktivnosti kojima je vreme podsećanja prošlo se prikazuju korisniku u posebnoj listi notifikacija za podsećanje.



Slika 7. Prikaz aktivnosti u kalendaru

6. ZAKLJUČAK

Najkomplikovaniji deo prilikom procesa razvoja poslovnog sistema jeste projektovanje modela podataka. Da bi se ovaj korak kvalitetno realizovao, potrebno je prikupiti veliku količinu domenskog znanja kako bi se kreirala arhitektura koja pokriva sve slučajeve korišćenja. Sve veći broj predmeta, a samim time informacija i dokumenta, advokatima otežava posao. Čuvanje podataka u čistom tekstu na papiru usporava svakodnevni rad jer papir onemogućava jaku povezanost između dokumenata kao i laku pretragu.

Prednosti namenskog poslovnog sistema za advokate su samim tim neizmerne. Implementiran softver je od početka projektovan imajući u vidu pre svega advokatske kancelarije u Republici Srbiji i time ima veliku prednost s

obzirom na manjak konkurencije u oblasti. Uz dodatne konsultacije sa advokatima i naknadnu doradu, softver bi mogao da postane komercijalno rešenje za ove probleme. Očigledna mesta za unapređenja ovog softvera su napredno pretvaranje slika u tekst ili automatsko popunjavanje formi na osnovu postojećih papirnih dokumenata.

6. LITERATURA

- [1] <https://www.mpravde.gov.rs/cron/files/O/2018/O2018Y00.html> (pristupljeno u septembru 2019.)
- [2] <https://en.wikipedia.org/wiki/JasperReports> (pristupljeno u septembru 2019.)
- [3] <https://www.elastic.co/what-is/elasticsearch> (pristupljeno u septembru 2019.)
- [4] <https://existek.com/blog/top-front-end-frameworks-2019/> (pristupljeno u septembru 2019.)
- [5] <https://www.paragraf.rs/propisi/sudskiposlovnik-2.htm> (pristupljeno u septembru 2019.)
- [6] <https://www.oodesign.com/factory-pattern.html> (pristupljeno u septembru 2019.)

Kratka biografija:



Ilija Grbić rođen je u Novom Sadu 1996. god
Diplomski rad na Fakultetu tehničkih nauka iz
oblasti Elektrotehnike i računarstva –
Biblioteka za neuronske mreže u
programskom jeziku C++ odbranio je
2018.god. kontakt: ilijagrbić@uns.ac.rs