I Faza projekta iz Arhitekture i projektovanja softvera

Arhitektura sistema

TaskIT

Marija Trajković, 18020

Elektronski fakultet, Niš

1. *Kontekst i cilj softverskog projekta*

TaskIT je višekorisnička web aplikacija, njena svrha je povezivanje korisnika koji mogu da postavljaju različite poslove koje žele da neko drugi obavi umesto njih i korisnike koji žele da te poslove obavljaju. Korisnik može da se prijavi na svoj nalog, da postavi posao koji želi da neko obavi za njega, da se i sam prijavljuje za poslove, da otkazuje poslove, da se pretplati na tipove poslova koje bi obavljao ili na korisnike za koje bi oobavljao te poslove i dobija obaveštenja.

Cilj aplikacije je da oslobodi ljude neželjenih poslova , a za koje su spremni da plate nekoga kome bi ta izrada bila korisna, najčešće novčano.

1. *Arhitekturno specifični zahtevi, glavni funkcionalni zahtevi I ne-funkcionalni zahtevi (atribute kvaliteta), tehnička i poslovna ograničenja*
   1. Glavni funkcionalni zahtevi
      1. Kreiranje korisničkog naloga (Registracija)
      2. Logovanje na nalog (Prijavljivanje)
      3. Kreiranje oglasa posla
      4. Izmena oglasa za posao
      5. Brisanje oglasa za posao
      6. Pregled dostupnih oglasa za posao
      7. Praćenje poslova
      8. Praćenje korisnika
      9. Prijavljivanje za obavljanje posla
      10. Otkazivanje prijave za neki posao
      11. Obaveštavanje o prijavi za obavljanje posla i o ponovo dostupnim poslovima
      12. Skladištenje podataka(o korisnicima, poslovima)
   2. Ne-funkcionalni zahtevi
      1. Pouzdanost i dostupnost (Reliability, Availability)– sistem treba da omogući perzistentnost akcija u konačnom vremenskom periodu, da bude dostupan korisnicima u svakom trenutku, kao i da bude otporan na otkaze i greške.
      2. Modifikabilnost (Modifiability) – sistem treba organizovati na način da on bude lako proširljiv bez potrebe za većim izmenama u sistemu u toku njegovog životnog veka, a radi zadovoljenja novih (ne -) funkcionalnih zahteva.
      3. Skalabilnost (Scalability) – sistem može da podrži rast broja konkurentnih korisnika.
      4. Performanse (Performance) – sistem ima što manje vreme odziva kako bi se ispunio uslov real-time aplikacije.
      5. Sigurnost (Security) – sistem treba da garantuje bezbednost podataka korisnika od neovlašćenog pristupa i zloupotrebe podataka.
      6. Upotrebljivost (Usability)- sistem treba da bude jednostavan i intuitivan za širok spektar korisnika.
      7. Testabilnost (Testability) – sistem je jednostavnog dizajna i moguće ga je lako testirati.
   3. Tehnička I poslovna ograničenja
      1. Kako je sistem web aplikacija, neophodno je korišćenje web tehnologija koje omogućuju pokretanje aplikacije u različitim web pretraživačima i jednostavnu i efikasnu komunikaciju korisnika i sistema.
      2. Implementacija modela podataka i njihova reprezentacija u bazi podataka je sakrivena od korisnika aplikacije.
      3. Komunikacija je podržana kroz sinhroni i asinhroni tip komunikacije. Sinhroni tip komunikacije se odvija između klijenta i servera, dok se asinhroni tip komunikacije odvija između samih klijenata prilikom postavljanja, brisanja i otkazivanja poslova.
2. *Arhitekturni dizajn softverskog sistem*

* Arhitekturni obrasci koji će biti korišćeni

1. Layered obrazac – sistem će biti realizovan u troslojnoj arhitekturi, sa sledećim slojevima: sloj perzistencije , serverski sloj i prezentacioni klijentski sloj. Sloj perzistencije je u osnovi, komunicira sa serverskim slojem i omogućava skladištenje podataka tj. predstavlja samu bazu podataka.Prezentacioni sloj obezbeđuje interakciju korisnika sa sistemom preko korisničkog interfejsa. Povezan je sa serverskim slojem. Serverski sloj predstavlja vezu između sloja perzistencije i prezentacionog sloja. Izvršava se na serveru i implementira poslovnu logiku sistema, funkcije za persistenciju podataka i sinhronu i asinhronu komunikaciju sa klijentom.
2. Model-View-Controller obrazac -
3. Command obrazac-
4. Observer obrazac – sistem ovaj obrazac implementira kod obaveštavanja korisnika o dostupnosti

* Generalna arhitektura (box-line model)
* Dijagrami osnovnih modula (komponenti) i njihovih veza (konektora)

1. *Aplikacioni okvir(i)*

Od tehnologija, koristiću:

* 1. React na frontendu,
  2. .NET na backendu,
  3. za ostvarivanje real-time komunikacije SignalR.