

Elektrotehnički fakultet, Univerzitet u Beogradu

SI3PSI Principi Softverskog Inženjerstva

Projekat
Majstor

Verzija 1.1

Tim “Nedeljive muve”

David Milinković 2020/0037

Mihailo Timotić 2020/0047

Stefan Bukorović 2020/0111

Pavle Janevski 2020/0055

1. Uvod	3
1.1. Rezime	3
1.2. Namena dokumenta	3
2. Opis problema	3
3. Kategorije korisnika	3
3.1. Registrovan korisnik	4
3.2. Administrator	4
4. Opis proizvoda	4
4.1. Pregled arhitekture sistema	4
4.2. Pregled karakteristika	5
4.3. Postojeće aplikacije na tržištu	5
5. Funkcionalni zahtevi	6
5.1. Registracija korisnika	6
5.2. Administriranje sistema	6
5.3. Kreiranje oglasa	6
5.3.1. Unošenje naslova oglasa	6
5.3.2. Unošenje opisa problema	6
5.3.3. Unošenje slika	7
5.3.4. Unošenje kategorija veština	7
5.3.5. Unošenje očekivane naplate	7
5.4. Prijava na oglas	7
5.4.1. Unošenje cene	7
5.4.2. Unošenje ponude	7
5.5. Ocenjivanje i recenzija	8
5.5.1. Verifikacija obavljenog posla	8
5.5.2. Unošenje ocene i komentara	8
5.6. Prihvatanje ponude za oglas	8
5.6.1. Odabir ponude	8
5.7. Razgovor između majstora i klijenta	9
5.7.1. Slanje poruka	9
5.7.2. Pravljenje konačnog dogovora	9
6. Pretpostavke i ograničenja	9
7. Kvalitet	9
8. Nefunkcionalni zahtevi	9
8.1. Sistemski zahtevi	9
8.2. Ostali zahtevi	10
9. Zahtevi za korisničkom dokumentacijom	10
9.1. Uslovi korišćenja	10
9.2. Uputstvo za administratore	10

1. Uvod

1.1. Rezime

Projekat Majstor je deo praktične nastave na predmetu Principi softverskog inženjerstva na 3. godini osnovnih studija. Cilj izrade projekta je upoznavanje sa svim fazama izrade većeg timskog softverskog projekta od ideje do implementacije i testiranja.

1.2. Namena dokumenta

Ovaj dokument napravljen je sa ciljem opisivanja idejnog rešenja, osnovnih funkcionalnosti i specifikacija aplikacije. Namenjen je za čitanje od strane svih članova tima kao i za klijenta za koga se aplikacija pravi.

2. Opis problema

Aplikacija Majstor bi trebalo korisnicima da reši problem efikasnog pronalaženja majstora za kućne poslove. Pored ovog, aplikacija treba da omogući lakši i efikasniji način majstorima da nađu posao. Korisnici bi mogli da postavljaju probleme koje imaju u kući uz opis problema i slike, na koji bi majstori mogli da se prijave sa predloženom cenom i terminom. Korisnik koji je postavio problem bi mogao posle toga od više ponuda da izabere onu koja mu najviše odgovara. Sistem bi takođe implementirao mehanizme ocenjivanja i poverenja, uz opcije da se dolazak majstora na lice mesta verifikuje skeniranjem QR koda od strane korisnika pomoću aplikacije, obavljeni posao oceni i potencijalno prijavi nestručno i nesavesno ponašanje majstora, kao i mnoge druge funkcionalnosti koje bi bile implementirane uz posredstvo administratorskog tima sistema i koje bi omogućile pouzdan i siguran način da se pronađe odgovarajući majstor.

3. Kategorije korisnika

U aplikaciji Majstor razlikuje nekoliko vrsta korisnika:

- Registrovan korisnik
 - Uloga klijenta
 - Uloga majstora
- Administrator

3.1. Registrovan korisnik

Registrovan korisnik može da vrši ulogu klijenta ili majstora. Klijent može da postavi oglas sa opisom problema koji ima. Majstor odgovara na oglas koji je klijent postavio i šalje svoju ponudu za dati posao. Klijent može odabrati majstora od onih koji su se javili na oglas ili direktno kontaktirati željenog majstora. Klijent može oceniti rad majstora sa mogućnošću pisanja dodatnih komentara o njegovom radu. Majstor može oceniti i ostaviti komentar o klijentu.

3.2. Administrator

Administrator ima mogućnost da postavi ili ukloni naloge ili naloge korisnika ukoliko za to ima potrebe, kao i oglase koji krše pravila aplikacije.

4. Opis proizvoda

U ovoj sekciji opisani su osnovni pojmovi od značaja za sistem. Dat je pregled arhitekture sistema na najvišem nivou i sumirane su glavne karakteristike sa stanovišta pogodnosti za korisnike.

4.1. Pregled arhitekture sistema

Sistem je zamišljen kao dinamička web aplikacija koja se sastoji od:

- MySQL baze podataka u kojoj se perzistiraju svi potrebni podaci za funkcionisanje sistema (podaci korisnika, istorije popravki, ocene...)
- Serverske aplikacije (kreirane pomoću Django framework-a) koja ima pristup gorepomenutoj bazi podataka. Ova aplikacija spoljnim sistemima pruža REST API pomoću kojeg je moguće pribavljati podatke i izvršavati korisničke akcije. Krajnje tačke ovog API-a su obezbeđene tako da određene akcije mogu da izvršavaju samo određeni tipovi korisnika, tj. pre obrade svakog zahteva vrše se autentifikacija i autorizacija.
- Korisničke jednostranične Web aplikacije (*engl: Single-page Web App*) (kreirane uz pomoć React JavaScript framework-a) koja pomoću slanja asinhronih HTTP zahteva (Ajax) ka gorepomenutoj serverskoj aplikaciji pribavlja sve potrebne podatke i omogućuje izvršavanje akcija korisnika. Ovu aplikaciju koriste svi navedeni tipovi korisnika, a korisnički interfejs, tj. dostupne funkcionalnosti, variraju u zavisnosti od tipa prijavljenog korisnika.

4.2. Pregled karakteristika

U datoj tabeli ispod navedene su najvažnije arhitekturne karakteristike koje bi sistem trebalo da ispuni.

Karakteristika	Obrazloženje
Skalabilnost	Sistem nije namenjen za određenu grupaciju ili kompaniju, pa se može očekivati da broj korisnika bude značajno veliki. Prilikom izrade projekta, potrebno je težiti da sistem bude takav da je jednostavno promeniti hardversku arhitekturu i/ili distribuirati ga na više fizički raspregnutih servera, a da se pritom očuvaju sve funkcionalnosti.
Modularnost	Potrebno je težiti tome da sistem bude što modularniji, tj. da je sastavljen od dosta slabo spregnutih komponenata, što će imati brojne benefite u pogledu jednostavnosti testiranja, jednostavnosti razvoja od strane više programera, jednostavnosti procesa izmene funkcionalnosti ili uvođenja novih funkcionalnosti...
Razumljivost	Korisnici sistema biće osobe širokog spektra upoznatosti sa korišćenjem Web aplikacija i interneta generalno, što nameće potrebu za intuitivnim i jednostavnim korisničkim interfejsom, funkcionalnostima koje su jasno definisane i koje imaju kratke i koncizne opise... Korisnicima ne bi trebalo da je potrebna obuka za korišćenje sistema. Takođe, korisnički interfejs trebalo bi da bude podjednako razumljiv i na mobilnim uređajima i na računarima, sa nešto većim akcentom na mobilne uređaje, zbog potrebe za fotografisanjem u okviru pojedinih funkcionalnosti, potrebe za portabilnošću proizašle iz činjenice da majstor mora napustiti svoj dom a pritom i dalje koristiti aplikaciju...

4.3. Postojeće aplikacije na tržištu

Na tržištu postoji svega par aplikacija koje samo delimično ispunjavaju zahteve korisnika.

Aplikacija NadjiMajstora, iako idejno dosta slična, ne ispunjava sve zahvete tržišta pa je to i njen glavni nedostatak. Aplikacija nudi jednosmeru komunikaciju gde samo klijent može da kontaktira majstora, ali ne i obrnuto. Ovo za posledicu ima da majstori koji nemaju veliki broj ocena ne dolaze do izražaja, pa nemaju ni podsticaj da se prijavljuju na sajt i aktivno učestvuju. Dodatno, majstori sa velikim brojem recenzija su često

prebukirani, što dovodi do dugog vremena čekanja na uslugu i samim tim klijenti gube želju za korišćenjem sajta.

Aplikacija Majstor teži da upravo ove nedostatke nadomesti i omogućiti ne samo jednokratno rešenje za klijenta, već dugoročno korišćenje aplikacije.

5. Funkcionalni zahtevi

U ovom odeljku definišu se osnovne funkcije koje sistem treba da obezbedi različitim kategorijama korisnika.

5.1. Registracija korisnika

Ukoliko korisnik nema svoj nalog, može ga kreirati unošenjem ličnih podataka. Ti podaci će biti upisani u bazu podataka na osnovu čega će kasnije moći da pristupa sistemu. Korisnik će morati da potvrdi svoj nalog preko e-mail adrese koju je uneo prilikom registracije.

5.2. Administriranje sistema

Administrator sistema ima najveća ovlašćenja, gde može da odobrava zahteve za drugim administratorskim nalogom. Administrator je u takođe mogućnosti da briše naloge/oglase na osnovu prijave.

5.3. Kreiranje oglasa

Svi korisnici sistema mogu kreirati oglas. Unos potrebnih podataka za oglas se vrši preko HTML forme gde je potrebno uneti sledeće podatke

5.3.1. Unošenje naslova oglasa

Potrebno je uneti kratak tekst koji bi predstavljao naslov oglasa, nešto po čemu bi se oglas mogao lako pretražiti

5.3.2. Unošenje opisa problema

Ovde je potrebno malo detaljnije opisati problem koji korisnik ima, opisati kako kvar izgleda i kada se dešava.

5.3.3. Unošenje slika

Moguće je uneti nekoliko slika koje mogu lepo da dočaraju majstoru kakav je kvar u pitanju, kako bi se kasnije ubrzala komunikacija oko tačnog problema koji je nastao.

5.3.4. Unošenje kategorija veština

Od ponuđenih kategorija za veštine majstora, potrebno je odabrati one za koje korisnik misli da najbolje odgovaraju datom problemu. Na primer, za problem sa bojlerom biće moguće odabrati kategoriju sa nazivom vodoinstalater ili nešto slično.

5.3.5. Unošenje očekivane naplate

Potrebno je uneti koliko bi dati korisnik želeo da plati za rešavanje ovog problema. Ovo predstavlja samo okvirnu cenu, kasnije će majstor i korisnik moći da se dogovore oko cene detaljnije.

5.4. Prijava na oglas

5.4.1. Unošenje cene

Majstor treba da unese cenu koja predstavlja njegovu ponudu za zadati problem.

5.4.2. Unošenje ponude

Potrebno je uneti tekst gde majstor može da se predstavi, kaže šta misli o problemu, da neku procenu potrebnog rada/cene.

5.4.3. Slanje ponude

Klikom na dugme šalje se ponuda klijentu prilikom čega se kreira i odeljak za komunikaciju između klijenta i majstora vezano za taj postavljeni problem.

5.5. Ocenjivanje i recenzija

5.5.1. Verifikacija obavljenog posla

Kako bi se sprečila zloupotreba sistema poverenja u slučaju ocenjivanja majstora od strane korisnika koji uopšte nisu imali zahtev koji je traženi majstor ispunio i zatim bio ocenjen (tzv. bratsko ocenjivanje), sistem verifikacije bi donekle mogao sprečiti ovu pojavu. Od majstora i klijenta bi se očekivalo da fotografišu subjekat klijentovog zahteva pre i nakon popravke, a kako bi se verifikovalo da su majstor i klijent stvarno bili fizički istovremeno na istom mestu, mogao bi se koristiti mehanizam verifikacije pomoću QR koda - majstor (ili klijent) u procesu verifikacije na ekranu mobilnog uređaja dobija QR kod koji se menja na svakih nekoliko sekundi i čiji sadržaj generiše i pamti serverska aplikacija; klijent (ili majstor) skenira ovaj QR kod, i time se donekle uspostavlja verifikacija fizičkog susreta klijenta i majstora. Naravno, ovo nije idealno rešenje i bilo bi podložno manipulacijama, ali bi broj istih bio dosta manji u odnosu na sistem bez ovakvog vida zaštite.

5.5.2. Unošenje ocene i komentara

Po obavljenoj verifikaciji, klijent može oceniti rad majstora ocenom od 1 do 5 kao i ostaviti komentar na rad majstora. Ocene i komentari su vidljivi svim korisnicima.

5.6. Prihvatanje ponude za oglas

5.6.1. Odabir ponude

Klijentu se prikazuju sve ponude poslate od strane majstora. Za one ponude koja ne odgovaraju klijentu klijent može odbiti datu ponudu. Za one o kojima bi želeo detaljnije da se dogovori, može klikom na dugme Prihvati ući u konverzaciju sa datim majstorom gde se mogu dogovoriti oko detalja.

5.7. Razgovor između majstora i klijenta

5.7.1. Slanje poruka

Majstor i klijent se mogu porukama dogovoriti oko finalnog termina. Dok se dopisuju i napreduju kroz dogovor izlaze obaveštenja o tome u kojoj fazi se nalaze sa dogovorom.

5.7.2. Pravljenje konačnog dogovora

Nakon što su se klijent i majstor dogovorili, klijent može predložiti da se dogovor zaključa, odnosno da se pređe u finalizaciju dogovora. Nakon što majstor to prihvati, ostaje da se završi i verifikuje posao.

6. Pretpostavke i ograničenja

Potrebno je težiti jedinstvenom dizajnu čitavog sajta. Oglasi za različite vrste poslova treba da poštuju istu formu.

Ne postoje ograničenja u funkcionalnostima koje klijenti i majstori mogu obavljati. Iz pogleda sistema, vidi se samo korisnik, koji ili postavlja oglase ili na njih odgovara. Ovo omogućava da majstor istovremeno može i postavljati oglase za sopstvene probleme.

7. Kvalitet

Potrebno je izvršiti testiranje metodom crne kutije svih gore navedenih funkcionalnosti. Potrebno je izvršiti testiranje kapaciteta i brzine odziva sistema, a takođe proveriti i konzistentnost podataka u bazi.

8. Nefunkcionalni zahtevi

8.1. Sistemski zahtevi

Serverska aplikacija bi se izvršavala na virtuelnoj/fizičkoj mašini koja ima instaliran sav potreban softver za izvršavanje Python programa, ima dostupnu internet konekciju i javno pristupljivu statičku IP adresu. Web aplikacija morala bi da bude podjednako upotrebljiva u svim modernim Web pregledačima, kako na desktop računarima tako i na mobilnim uređajima.

8.2. Ostali zahtevi

Kako biznis model sistema nema definisan vremenski period u kojem bi se sistem koristio, dostupnost sistema morala bi da bude veoma blizu 100%, a svaki prekid u radu radi održavanja sistema morao bi biti blagovremeno najavljen korisnicima, zbog specifičnosti sistema ocenjivanja i njegove važnosti za poverenje korisnika (verifikacija ne može da se odloži - klijent i majstor verifikaciju vrše istovremeno sa iste lokacije).

9. Zahtevi za korisničkom dokumentacijom

9.1. Uslovi korišćenja

Usaglašavanjem sa uslovima korišćenja, korisnici se obavezuju na savesno korišćenje sajta i oslobađaju vlasnike sajta odgovornosti u slučaju problema između dva korisnika.

9.2. Uputstvo za administratore

Uputstvo za administratore zadrži instrukcije za rad administratora. Instrukcije se sastoje od smernica u vezi moderacije oglasa (šta oglasi smeju ili ne smeju sadržati, kakav format moraju poštovati, ...) kao i korake koje administratori treba da prate u slučaju prijave nekog problema od strane korisnika

10. Plan i prioriteti

Primarno, u prvoj razvojnoj verziji potrebno je obezbediti sledeće funkcionalnosti:

- Autorizovanje korisnika
- Prijava korisnika na sistem
- Kreiranje oglasa
- Kreiranje ponude

U narednim fazama razvoja planirano je dodavanje pomenutih funkcionalnosti kao što su:

- Komunikacija između korisnika
- Ocenjivanje i postavljanje komentara
- Verifikacija urađenog posla korišćenjem QR koda i drugih metoda
- Dohvaćanje informacija o korisnicima

Nakon ovih, moguće je dodati i funkcionalnosti za pregled različitih statističkih podataka u aplikaciji.