
Elektrotehnički fakultet u Beogradu
Katedra za računarsku tehniku i informatiku

Predmet: Testiranje softvera (SI3TS / 13S113TS)

Nastavnik: Prof. dr Dragan Bojić

Asistent: Doc. dr Dražen Drašković

Ispitni rok: Februar 2019.

Datum: 08.02.2019.

Kandidat:* _____

Broj indeksa:* _____

*Ispit traje 2.5 sata, a u toku prvog sata nije dozvoljeno napuštanje ispita.
Upotreba literature nije dozvoljena.*

<i>Zadatak 1</i>	_____ /12	<i>Zadatak 4</i>	_____ /10
<i>Zadatak 2</i>	_____ /6	<i>Zadatak 5</i>	_____ /16
<i>Zadatak 3</i>	_____ /6	<i>Zadatak 6</i>	_____ /10

Ukupno na ispitu: _____ /60 *Ukupno na domaćem*:* _____ /40

Rok u kome je odbranjen domaći:* _____ (primer: januar 2019)

Ukupno: _____ /100

Ocena: _____ (_____)

Napomena: Ukoliko u zadatku nešto nije dovoljno precizno definisano, student treba da uvede razumnu pretpostavku, da je u okviru (da bi se lakše prepoznala prilikom ocenjivanja) i da nastavi da izgrađuje preostali deo svog odgovora na temeljima uvedene pretpostavke. Kod pitanja koja imaju ponuđene odgovore treba **samo zaokružiti** jedan odgovor. Na ostala pitanja odgovarati **čitko, kratko i precizno**. * popunjava student.

NAPOMENA: Programski kod u prvom zadatku i tekst zadatka (pre koda) je zajednički za zadatke 1, 2 i 3.

1. [12] Dat je C++ program koji broji i ispisuje broj printabilnih znakova, reči i linija u ulaznom tekstu. Reči su razdvojene blankom ' ' ili tabom '\t'(nisu printabilni znaci), a linije znakom kraja linije '\n'. Tekst se završava EOF znakom. Napomena: ne razmatrati u testiranju situacije da nema EOFa na ulazu, on je uvek prisutan.

```
1. #define YES 1
2. #define NO 0
3. main()
4. {
5.     int c, nl, nw, nc, inword;
6.     inword = NO;
7.     nl = 0;
8.     nw = 0;
9.     nc = 0;
10.    cin >> c;
11.    while (c != EOF) {
12.        nc = nc + 1;
13.        if (c == '\n')
14.            nl = nl + 1;
15.        if (c == ' ' || c == '\n' || c == '\t')
16.            inword = NO;
17.        else if (inword == NO) {
18.            inword = YES;
19.            nw = nw + 1;
20.        }
21.        cin >> c;
22.    }
23.    cout << nl << endl;
24.    cout << nw << endl;
25.    cout << nc << endl;
26. }
```

- a) Razviti skup testova analizom graničnih vrednosti.
- b) Navesti potencijalne nedoslednosti i nedorečenosti opisa programa u odnosu na implementaciju i dopuniti/korigovati opis da se slaže sa implementacijom.
- c) Opisati ulaz programa koristeći BNF ili EBNF notaciju.

2. [6] Razviti testove koristeći preporuke za testiranje petlji.

3. [6] Odrediti broj ciklomske kompleksnosti programa i *McCabe*-ovim metodom razviti skup testova.

4. [10] Dat je sledeći programski kod na programskom jeziku C:

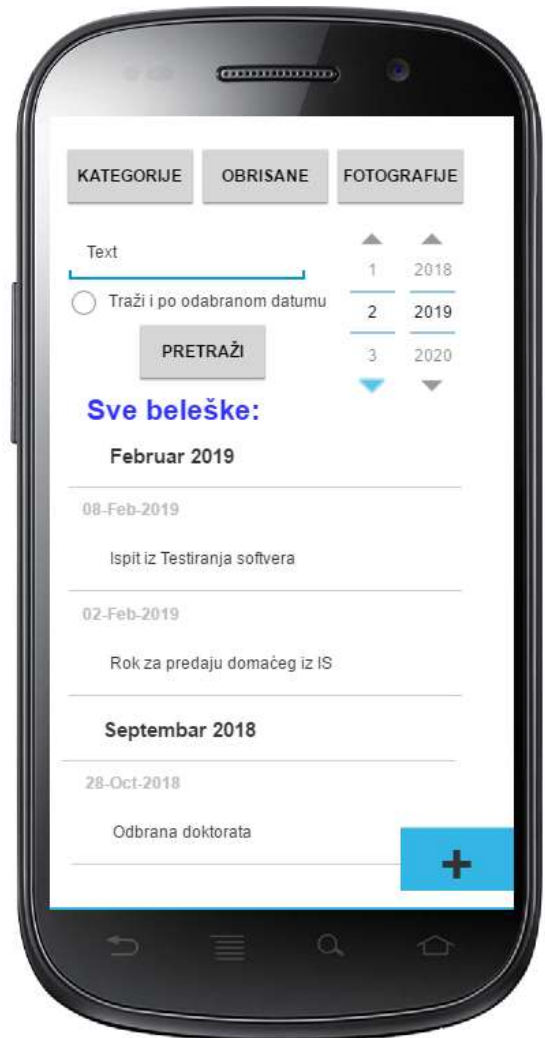
```
1. #include<stdio.h>
2. int main() {
3.     int n, first = 0, second = 1,
4.         next, c;
5.     printf("Enter the number of terms\n");
6.     scanf("%d",&n);
7.     printf("First %d terms of Fibonacci series are :-\n",n);
8.     for ( c = 0 ; c < n ; c++ )
9.     {
10.        if ( c <= 1 )
11.            next = c;
12.        else
13.        {
14.            next = first + second;
15.            first = second;
16.            second = next;
17.        }
18.        printf("%d\n",next);
19.    }
20.    return 0;
21. }
```

- a) [5] Naznačiti sve definicije, C- i P-upotrebe, kao i DU lance, koje je moguće formirati. Nakon toga prikazati minimalni skup test primera koji pokriva sve navedene DU lance.
- b) [5] Napraviti sledeće mutacije prvog reda u datom programu:
- U liniji 8, uslov u FOR petlji se mutira u negaciju uslova.
 - U liniji 10 relacioni operator mutira u sve druge moguće relacione operatore.
 - Linija 15 postaje linija 16, a linija 16 dolazi na mesto linije 15 programskog koda.

Koliko se mutant programa dobija i kako oni utiču na mutacioni skor (odnosno da li su LIVE ili KILL), ako primenjujemo test primer(e) dobijen(e) u tački a)?

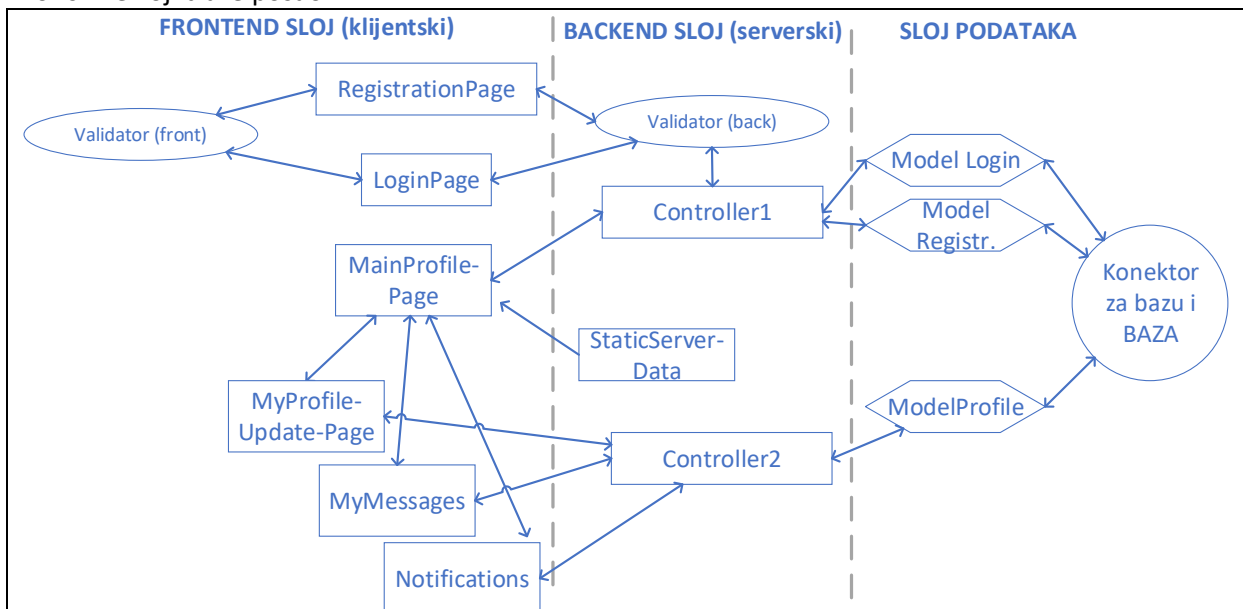
5. [16] Data je specifikacija mobilne aplikacije „Moja beležnica“ za prikupljanje različitih vrsta beleški. Pri učitavanju aplikacije prikazuju se sve beleške, poređanje opadajuće, od najnovijih ka najstarijim datumima. Za svaki mesec treba ispred beleški da se ispiše naziv meseca (kao na slici). Svaka beleška može pripadati nekoj kategoriji, koja je kreirana, podrazumevana kategorija je „Opšte beleške“. Glavni ekran aplikacije ima 3 dugmeta: 1) dugme KATEGORIJE kojim se prelazi na novi ekran i prikazuje lista svih formiranih kategorija za koje postoje beleške; kategorije koje imaju beleške imaju broj u zagradi npr [7] i u njih može da se uđe, a one koje imaju [0] kod imena ne postoje snimljene; 2) dugme OBRISANE, koji pokazuje listu svih obrisanih beleški (one se ne prikazuju na glavnom ekranu i prilikom pokretanja filtera); svaku obrisanu belešku, klikom na X pored nje, možete zauvek izbrisati iz memorije telefona; 3) dugme FOTOGRAFIJE, koji prikazuje samo beleške sa fotografijom, takođe vremenski uređene opadajuće. Klikom na neku belešku, prikazuju se njen kompletan sadržaj i pritiskom na X se prebacuje u listu obrisanih.

Ispod 3 dugmeta postoji i filter kojim se može vršiti pretraga postojećih beleški po sadržaju i/ili po mesecu i godini (podrazumevano se učitava tekući mesec 02/2019, ali treba omogućiti opseg pretrage od vremena prve do poslednje formirane beleške). Pretraga treba da omogući filtriranje po unetom tekstu (u okviru tekstualnog polja) bez datuma, ili po unetom datumu (ako tekst nije unet) ili i po tekstu i po datumu, za slučaj da je odabrano radio dugme ispod tekstualnog polja. Kada se klikne znak „+“ iskače kontrolni meni za formiranje: 1) Nove beleške; 2) Nove slike, 3) Novog podsetnika. Nova beleška, na današnji datum u datom vremenskom trenutku snimanja, treba da formira tekstualnu belešku sa opisom do 1000 karaktera. Pre snimanja beleške bira se i kategorija (*Sport, Fun and Parties, Meeting, Book,...*), ako se ne izabere ostaje opšta kategorija. Nova slika nas prebacuje na kameru telefona, i nakon uspešnog fotografisanja, beleži sliku pod trenutnim datumom i vremenom. Novi podsetnik treba da omogući snimanje kratke tekstualne poruke (do 80 karaktera), sa datumom i vremenom, koji korisnik odabere (u formatu DD-MM-YYYY, HH:NN, gde su oznake DD za dan u mesecu, MM za mesec, YYYY za godinu, HH za sate i NN za minute tog alarma). Alarm se beleži kao klasična beleška, kategorije „Alarm“ i u tom trenutku izbacuje notifikaciju na telefonu, kada se desi, a nakon notifikacije, po ulasku u našu testnu aplikaciju, pritiska se dugme OK da bi se alarm ugasio. Iz svakog stanja Nova Beleška/Slika/Podsetnik, dugmetom BACK na telefonu, može da se odustane, i vrati na glavni ekran naše mobilne aplikacije. Isto se dešava i ukoliko uđemo u neki podmeni, koji smo odabrali klikom na određeno dugme na glavnom ekranu.



- a) [8] Korišćenjem metoda klasa ekvivalencije, kombinovano sa metodom graničnih slučajeva (za one podatke koji su dati specifikacijom), testirati ovu mobilnu aplikaciju. Označiti koje su klase legalne, a koje nelegalne, i sa koliko minimalno test primera je neophodno testirati uspešne scenarije. Navesti takve testove.
- b) [8] Datu specifikaciju prevesti u dijagram stanja, prikazan iz ugla korisnika sistema. Na svakom prelazu označiti šta se postiže tom akcijom i za slučaj da postoji odgovor aplikacije navesti ga (Uneta nova beleška, i slično...). Izvesti tabelu prelaza za 0-switch i 1-switch nivo.

6. [10] Neka je dat sledeći dijagram koji prikazuje jednu veb aplikaciju društvene mreže za poslovne korisnike koji traže posao:



Opis dijagrama:

Registration Page- veb strana za registraciju novog korisnika

LoginPage – veb strana za logovanje korisnika na sistem

Validator (X) – validatori na klijentskoj i serverskoj strani (razne provere za polja forme)

Controler1 – glavna biznis logika aplikacije za rad sa korisnicima (registrovanje/logovanje)

Controler2 – pomocna biznis logika aplikacije, koja je važna kada se korisnik uloguje

Main Profile Page i ostale strane u Frontu – Profil korisnika i njegove veb strane (nakon logovanja)

Model X – serverski kod koji komunicira između logike aplikacije i Baze podataka

StaticServerData – statički podaci koji se dovlače sa servera (npr. LinkedIn logo i reklame)

a) [6] Korišćenjem integracionog testiranja, gledajući arhitekturu ovog veb sistema (smatrati da je modul svaki element ovog dijagrama) i znajući specifikaciju opisanu ispod dijagrama, odabрати jednu najpogodniju tehniku integracionog testiranja i objasniti kako bi se ona primenila na ovoj aplikaciji. Smatrati da je baza podataka jedan modul, odnosno da mora da se napravi jedinstvena na nivou cele aplikacije. Smatrati da inicijalna baza nema korisnike, odnosno da korisnik mora prvo da se registruje uspešno, da bi se ulogovao. Takođe, korisnik mora prvo da napuni svoj profil (ažurira ga), da bi mogao da šalje poruke za nalaženje posla, i da prima notifikacije od IT kompanija koje traže kadrove.

b) [4] Da li je moguće na neki način testirati korisnički interfejs aplikacije na klijentskim (frontend) veb stranama bez baze podataka i bez serverskog dela? Objasniti kako, ako je moguće, ili zašto nije moguće, kao i šta je minimalno potrebno da se testira klijentski deo.

