



UNIVERZITET U SARAJEVU  
ELEKTROTEHNIČKI FAKULTET SARAJEVO

# DOMAĆA ZADAĆA 2

## VERIFIKACIJA I VALIDACIJA SOFTVERA

**Student: Mašović Haris**

**Indeks: 17993**

**Odsjek: Računarstvo i Informatika**

**Datum:**

**05.12.2018**

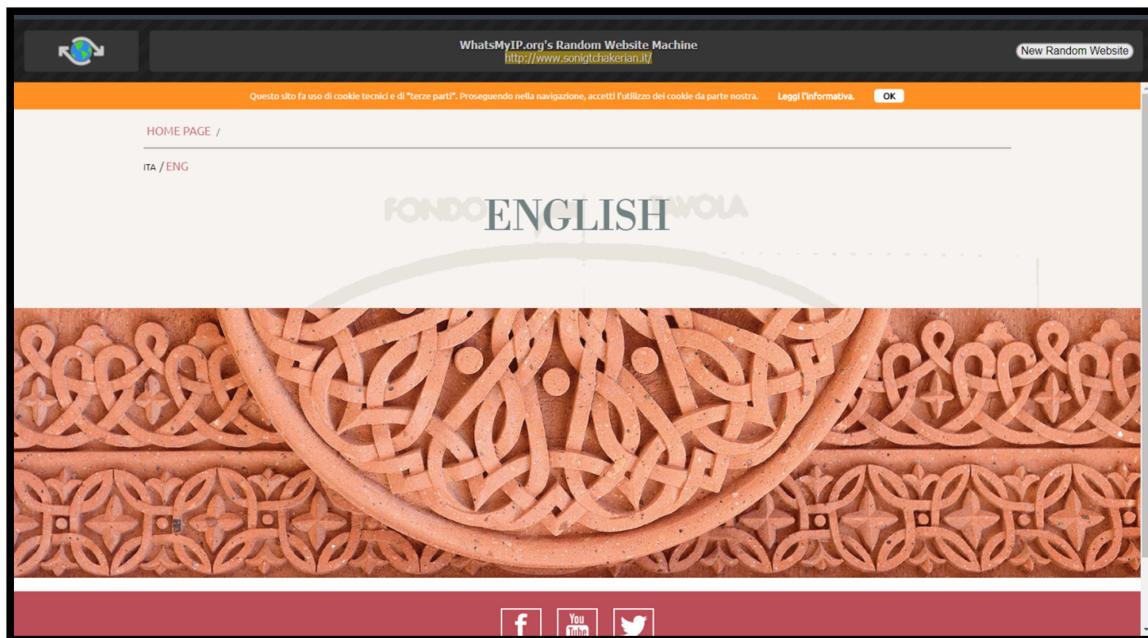
**Potpis:**

---

## ZADATAK 1: 2 boda

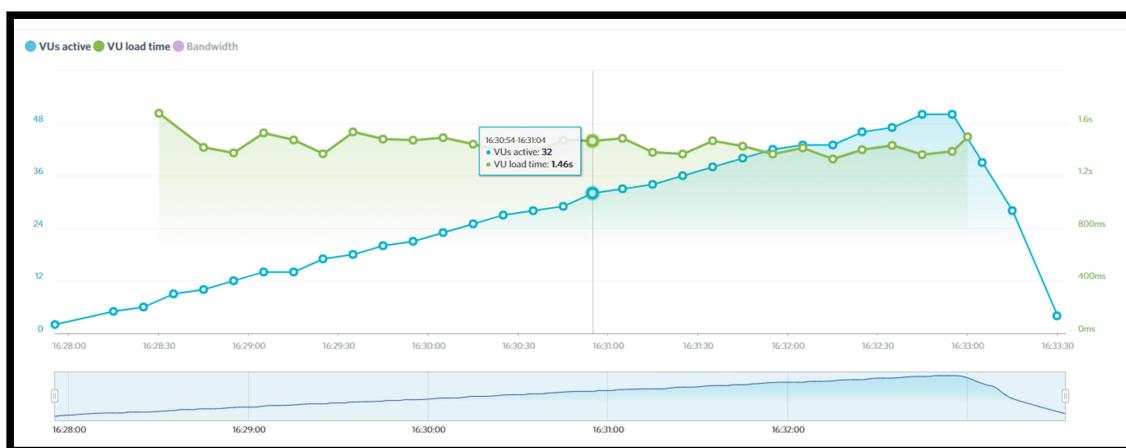
Za stranicu slučajno odabranu preko linka <http://www.whatsmyip.org/random-website-machine/> (priložite ekran dobijene stranice sa podacima kada je stranica odabrana, u slučaju dobijanja stranice sa neprikladnim sadržajem odabratи dugу stranicу) i sprovesti sljedeće testove :

Stranica koja je dobijena pomoću prethodnog linka je <http://www.sonigtchakerian.it/> odnosno po sljedećoj slici:



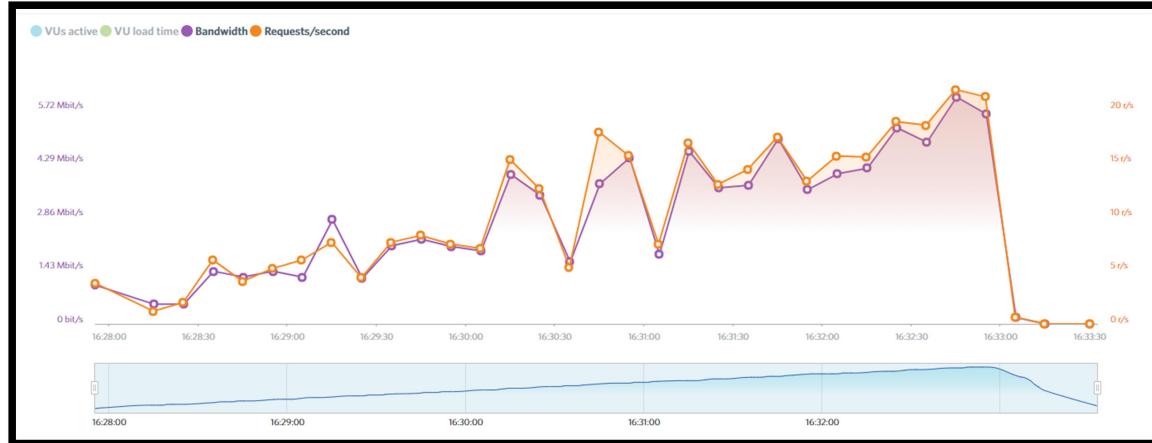
- Sprovesti kompletan scenario testiranja opterećenja koristeći alat *loadimpact*, te analizirati i dokumentirati dobijene rezultate. ( 0,5 bodova )

Specifirati ćemo da se testiranje održava sa 50 virtualnih korisnika i da se izvršava 5 minuta. Lokacija odakle se testira je Ashburn iz US. Nakon završetka izvršavanja opterećenja našeg dobijenog web sajta, imamo sljedeći rezultat.



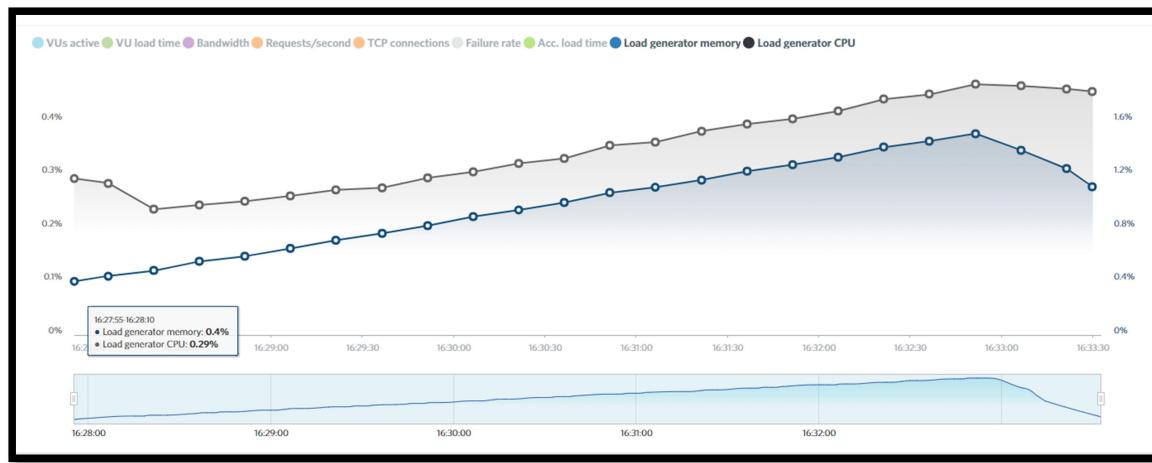
Sa prethodne slike vidimo graf koji nam prikazuje **poređenje odnosa broja aktivnih virtualnih korisnika i vremena koje je potrebno da se web aplikacija učita za jednog virtuelnog korisnika**. Sa slike vidimo da se generalno vrijeme učitavanja za jednog korisnika nakon određenog vremena stabiliziralo na 1.4s što nam ukazuje na relativno loše vrijeme odgovora za ovog korisnika. Ukoliko bi bio veći broj korisnika ovo vrijeme bi bilo veće. Naravno ovo nije pravi pokazatelj jer u realnim situacijama sigurno postoji mogućnost da više korisnika koristi stranicu.

Sljedeću analizu ćemo sprovesti na osnovu **poređenja bandwith-a u odnosu na broj requestova po sekundi**.



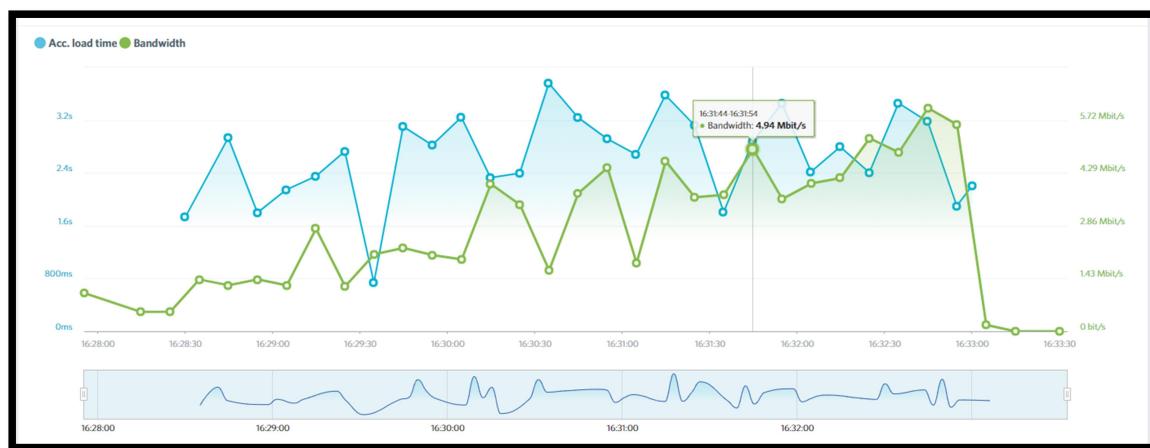
Sa našeg grafika možemo vidjeti da se broj requestova i propusnost ponašaju slično odnosno da u zavisnosti koliko korisnika pristupa sajtu da se propusnost povećava odnosno smanjuje. Vidimo da za 20 requestova po sekundi imamo najveću memorijsku propusnost što je i očekivano. Propusnost predstavlja ključan faktor u samoj opterećenosti stranice, shodno tome generalno je bolje imati što veću propusnost, a što manje kašnjenje.

Sljedeću analizu ćemo sprovesti na osnovu **poređenja zauzeća procenta iskorištenja cpu u odnosu na memoriju**.



Vidimo da se prilikom povećanja broja virtualnih korisnika, povećava zauzetost procesora u odnosu na memoriju, također se vidi da veličine zauzetosti su odlične za naše testiranje što se također podudara sa logikom, jer naša stranica nije toliko funkcionalno urađena ili da zahtijeva korištenje veće memorije u odnosu na standard. Naravno što manja zauzetost cpu/memorije znači efikasniju stranicu odnosno manju opterećenost.

Sljedeću analizu ćemo sprovesti na osnovu **propusnosti u vrijeme loadanja naše stranice**.



Iz ove analize možemo zaključiti da se pri slaboj propusnosti stranice sporije load-a odnosno razlog za više vremensko loadanje predstavlja manja vrijednosti propusnosti što smo dokazali analizom gore. Također ukoliko imamo preveliko kašnjenje to također može uticati na učitavanje stranice.

Vidimo na našem grafiku da se prilikom povećavanja bandwitha odnosno propusnosti, peek odnosno vrijeme loadanja smanjuje u zavisnosti od propusnosti, tj. povećavanjem propusnosti u određenim tačkama dobijamo manje vrijeme loadanja stranice.

Veća propusnost, a i manje kašnjenje imaju velik uticaj na samo loadanje stranice.

Na kraju možemo uraditi analizu naših API requestova odosno HTTP requestova.

URL	Load zone User scenario	Method	Status	Count	Size Compressed	Min Max	Avg
<a href="http://www.sonigtchakerian.it/..../font-awesome.css">www.sonigtchakerian.it/..../font-awesome.css</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	142	38.87 KB -	308ms 805ms	334ms
<a href="http://www.sonigtchakerian.it/..../font-awesome.css">www.sonigtchakerian.it/..../font-awesome.css</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	144	38.87 KB -	152ms 266ms	231ms
<a href="http://www.sonigtchakerian.it/..../install.css">www.sonigtchakerian.it/..../install.css</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	141	926 Bytes -	77ms 82ms	80ms
<a href="http://www.sonigtchakerian.it/..../install.css">www.sonigtchakerian.it/..../install.css</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	144	926 Bytes -	76ms 80ms	77ms
<a href="http://www.sonigtchakerian.it/..../install.css">www.sonigtchakerian.it/..../install.css</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	142	1.45 KB -	77ms 81ms	80ms
<a href="http://www.sonigtchakerian.it/..../install.css">www.sonigtchakerian.it/..../install.css</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	144	1.45 KB -	76ms 82ms	78ms
<a href="http://www.sonigtchakerian.it/..../stile.css">www.sonigtchakerian.it/..../stile.css</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	142	28.22 KB -	308ms 1.38s	340ms
<a href="http://www.sonigtchakerian.it/..../stile.css">www.sonigtchakerian.it/..../stile.css</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	144	28.22 KB -	228ms 1.38s	311ms
<a href="http://www.sonigtchakerian.it/en">www.sonigtchakerian.it/en</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	142	2.32 KB -	160ms 891ms	178ms
<a href="http://www.sonigtchakerian.it/en">www.sonigtchakerian.it/en</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	200 OK	146	2.32 KB -	158ms 337ms	166ms
<a href="http://www.sonigtchakerian.it/favicon.ico">www.sonigtchakerian.it/favicon.ico</a>	Ashburn, US (Amazon) Auto generated from www.sonigtchakerian.it	Get	404 Not Found	142	209 Bytes -	154ms 1.23s	189ms

Vidimo da za svaki request možemo vidjeti koja je to http metoda u pitanju, povratni status, broj metoda koje su pozvane za naše testiranje, veličinu podataka koji su se prenijeli, minimalno i maximalno vrijeme koje je bilo potrebno (uzima se rang jer imamo više requestova za isti API request) i naravno average tj. prosječno vrijeme koji je potrebno za naše requestove.

Može se zaključiti da average vrijeme zavisi od veličine fajla koji se traži. Najveće average vrijeme za neki request u našem testiranju je 983ms, dok najmanje 77ms. Ovo predstavljaju vremena za naše testiranje tj. za naš reprezentativni skup, pri čemu je logično da će se povećati ukoliko je veće opterećenje za stranicu. Naravno što manja veličina fajlova to znači bolje average vrijeme koje je potrebno da se prenesu fajlovi, što znači manja opterećenost stranice.

**Svi dosadašnji dijelovi analize predstavljaju uticaj na opterećenost stranice, što je i trebalo pokazati sa loadimpact alatom.**

**- Sprovedi kompletan scenario testiranja brzine odziva koristeći alat *pingdom*, te analizirati i dokumentirati dobijene rezultate. (0,5 bodova)**

Sprovedi ćemo testiranje na pingdom stranici sa lokacije Washington,USA. Kao rezultat testiranja imamo sljedeće:

Your Results:

Performance grade: C 77

Page size: 1.0 MB

Load time: 1.38 s

Requests: 15

DOWNLOAD HAR | SHARE RESULT

Vidimo da vrijeme potrebno za loadanje stranice je 1.38 sekunde, da prilikom loadanja imamo 15 requestova koji se moraju odraditi. Veličina stranice je 1MB, a sama performansa po pingdomu je 77, odnosno ocjena C što predstavlja na skali od 1-5 tricu.

Improve page performance

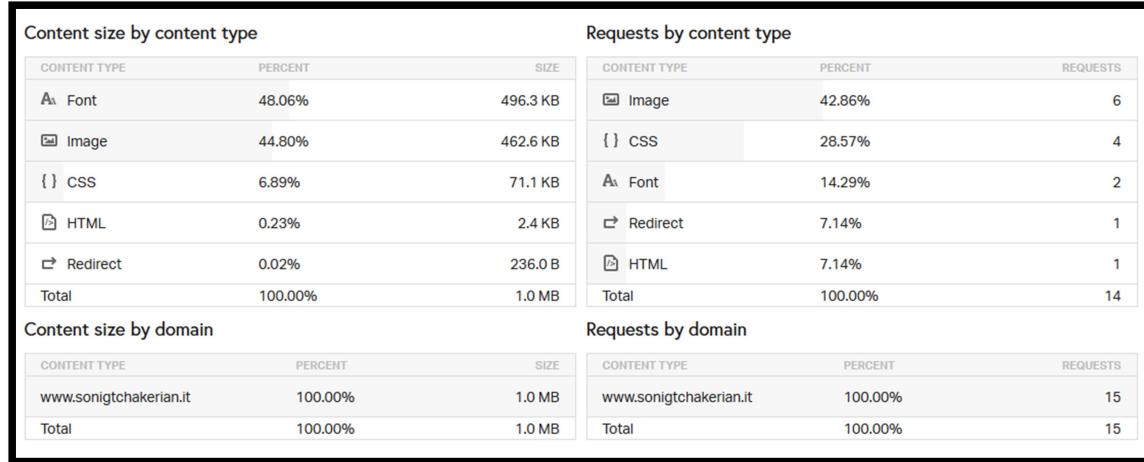
GRADE	SUGGESTION
F 0	Use a Content Delivery Network (CDN)
F 0	Configure entity tags (ETags)
F 0	Add Expires headers
F 45	Compress components with gzip
F 50	Use cookie-free domains
B 90	Avoid URL redirects
A 92	Make fewer HTTP requests

Response codes

RESPONSE CODE	RESPONSES
200 OK	13
302 Found	1
404 Not Found	1

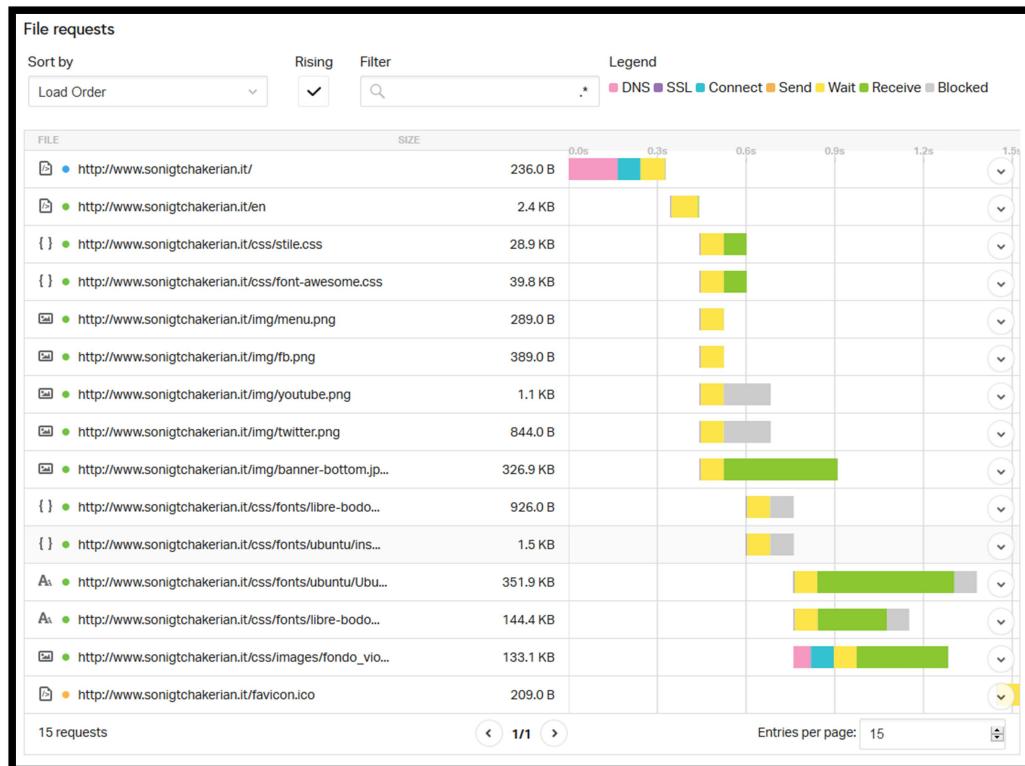
Za sljedeći dio analize, vidimo da se sastoji od 2 dijela. Prvi dio predstavlja '**Improve page performance**' odnosno načine na koji možemo povećati performanse naše stranice. Ovdje su prikazane razne strategije pomoću kojih možemo povećati performanse i pingdom zapravo mjeri da li se te strategije koriste. Isto kao i gore za performanse ocjena A predstavlja 5, B predstavlja 4 dok ocjena F predstavlja 1. Skor za svaku strategiju je prikazan desno pored ocjene.

Drugi dio predstavlja '**response codes**' odnosno predstavlja koje imamo response code-ove prilikom loadanja stranice. Vidimo da imamo 13 OK response-a, 1 Found response i 1 not found response. 2xx kod znači uspješan kod, dok 3xx kod predstavlja redirekciju, 4xx kod znači error na klijent strani, dok 5xx kod znači error na server strani.



Pretposljednja analiza se tiče mjere content-a i requestova po MIME tipu odnosno po samoj domeni. Vidimo da se svi requestovi i content nalaze na specificiranoj domeni što predstavlja drugi dio u našoj slici.

Prvi dio u našoj slici predstavlja rangiranje content-size-a i requestova po MIME tipu. Vidimo da po MIME tipu najviše ima Font-ova, slika, pa tek css-a dok generalno html i redirect vrijednosti ima relativno malo. Također vidimo na desnoj stranici slici, da što se tiče requestova najviše se vrše requestovi za slike i css fajlove, zatim za fontove redirekciju i html. Sami requestovi će biti opisani u posljednjoj fazi analize.



U posljednjem dijelu analize ćemo prikazati sve requestove. Za naš svaki request bilo kojeg MIME povratnog tipa sadrži veličinu, i vrijeme za koje se izvršava, unutar tog vremena može se izvršavati DNS potražanja SSL handshake, čekanje za neki drugi resurs, sama konekcija, prijem resursa i blokiranje.

Requestovi su poredani u redu pri kojem se loadala stranica.

**Svi dosadašnji nabrojani dijelovi analize utiču na brzinu naše stranice prilikom loadanja što je trebalo pokazati sa pingdom alatom.**

**- Odabratи dodatni алат за stress testiranja, првести testiranje и документирати добијене резултате. (1 бод).**

Izabrao sam <https://flood.io/> alat za stress testing. Veoma je sličan loadimpact alatu, te pogodan za prikaz rezultata stress testiranja. Prilikom ovog testiranja, vrši se sa 600 virtuelnih korisnika, i testira se iz Kalifornije tj. US West. Dodatne karakteristike stress testiranja su prikazane u sljedećoj slici. Vidimo da se testiranje vrši 5 minuta, i da se sa 600 virtuelnih korisnika testira stranica.

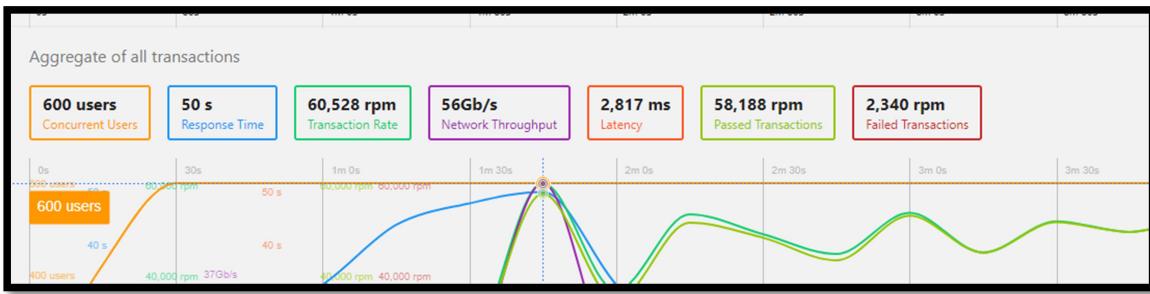
The screenshot shows the 'Configure Load and Launch' page of the Flood.io interface. At the top, it says 'Configure number of users, AWS regions, and test duration'. A note below states: 'For your initial tests, start with a fewer amount of users (1 to 500). Increase the load in subsequent tests.' The configuration section includes:

- 600 users per region**: A slider bar set to 600.
- AWS Regions**: Set to 'US West (California)'.
- Test Duration**: Set to 'Duration' of 5 Minutes and 'Ramp Up' of 0.5 Minutes.
- Summary**: Displays the test parameters: 'You are about to launch 600 users from 1 region for 5 minutes' and '0.20 Grid Node hours will be used from your account balance.'
- Buttons**: 'Cancel' and 'LAUNCH TEST' (highlighted in blue).

Rezultati testiranja su prikazani na sljedećoj slici:



Vidimo da se stalno prikazuje broj korisnika koji pristupa stranici, vrijeme odgovora, transakcijski rate odnosno broj rotacija, throughput tj. propusnost networka, broj transakcija koji su prošli u rotacijama i onih koji su 'pali'.



Kao primjer peek-a tj. najgoreg slučaja ovog stress testiranja imamo slučaj na polovini prve i druge minute. Vidimo da je aktivno 600 korisnika, da je za response time potrebno 50 sekundi, da se obrće oko 60k transakcija (requestova) pri čemu ima 58k transakcija koje su uspiješne i 2k transakcija koje nisu, da je propusnost mreže 56Gb/s, da je kašnjenje 2817ms odnosno 2.817s po requestu (pored velikog response time-a).

## ZADATAK 2: 3 boda

b) Demonstrirati *black box* testiranje - odabratи neku funkcionalnost aplikacije ( ukoliko ne odgovara sajt dobijen u zadatku 1 da se prikaže ova tehnika možete odabratи npr. projekat sa OOAD-a, projekat ili zadaću nekog drugog predmeta, ili neki drugi web sajt) i na osnovu dokumentacije, *he/pa* (ili izgleda ekrana) formirati prihvatljiv broj EC klasa. Na osnovu EC klasa formirati testne slučajeve. Izvršiti manuelno testiranje na osnovu testnih slučajeva (dovoljno za nekoliko testnih slučajeva) i sačiniti izvještaj o testiranju. ( 1 bod )

Moj dobiveni sajt u zadatku 1 ne odgovara da se prikaže ova tehnika, shodno tome odabrao sam RPR projekat i demonstraciju formiranja EC klasa za početnu formu programskog rješenja klinike. Pošto se generalno odradivala validacija u smislu provjeravanja username-a i passworda iz baze vezano za projekat, ovdje ćemo gledati zadovoljenost određenih parametara tj. validacija samog unosa pri čemu izvršiti modifikaciju koda, tako da ukoliko su zadovoljeni svi parametri da se loada neka druga proizvoljna forma odnosno u našem slučaju forma portirnice.

Sam izgled forme je dat na sljedećoj slici i dodatni parametri su sljedeći:

The screenshot shows a Windows application window titled "Registracija" for "Naša Malá Klinika". The window has a red cross icon in the center. It contains three text input fields with red borders: "Username:", "Password:", and "Potvrda Passworda:". Below these fields is a red "Prijava" button. At the bottom left, there is a green link "Opcije za pacijente".

- Da bi username bio zadovoljen mora se sastojati od niza karaktera i brojeva, pri čemu ukupna dužina ne smije biti manja od 3. Moguća su 2 ishoda, a to je da username jeste i nije validan,
- Da bi password bio zadovoljen mora se sastojati od malih slova (minilano 1 slovo) zatim tačke i niza brojeva pri čemu niz brojeva ne smije biti manji od 3, imamo 2 moguća ishoda: validno i nevalidno,
- Potvrda passworda je cross validacija za naš password i mora biti zadovoljen uslov jednakosti samih passworda u oba unosa, također su moguća 2 ishoda: validno i nevalidno.
- Vidimo da imamo 3 uslova na osnovu kojih ćemo ispitivati je li zadovoljen kriterij.

Za realizaciju same postavke iskoristiti ćemo novoformirani sljedeći kod:

```
bool provjeri = false;
var r = new Regex(@"^a-zA-Z0-9]{4,}$");
if (!r.IsMatch(korisnickoIme))
{
    toolStripStatusLabel1.ForeColor = Color.Red;
    toolStripStatusLabel1.Text = "Nije zadovoljen uslov za username! Unesite opet.";
    errorProvider1.SetError(textBox1, "Ne valja username.");
    provjeri = true;
}

var r2 = new Regex(@"^a-z]+\.[0-9]{3,}$");
if (!r2.IsMatch(sifra))
{
    toolStripStatusLabel1.ForeColor = Color.Red;
    toolStripStatusLabel1.Text = "Nije zadovoljen uslov za password! Unesite opet.";
    errorProvider1.SetError(textBox2, "Ne valja password.");
    provjeri = true;
}

if (!Validacije.validacijaPassworda(textBox2.Text, textBox3.Text))
{
    toolStripStatusLabel1.ForeColor = Color.Red;
    toolStripStatusLabel1.Text = "Ne poklapaju se šifre! Unesite opet potvrdu.";
    errorProvider1.SetError(textBox3, "Ne poklapa se šifra.");
    provjeri = true;
}
// ukoliko su svi uslovi zadovoljeni, loada se portirnica
if (!provjeri)
{
    Portirnica forma = new Portirnica(ref novaKlinika);
    this.Hide();
    forma.Show();
}
```

Formirajmo sada prihvatljive EC klase:

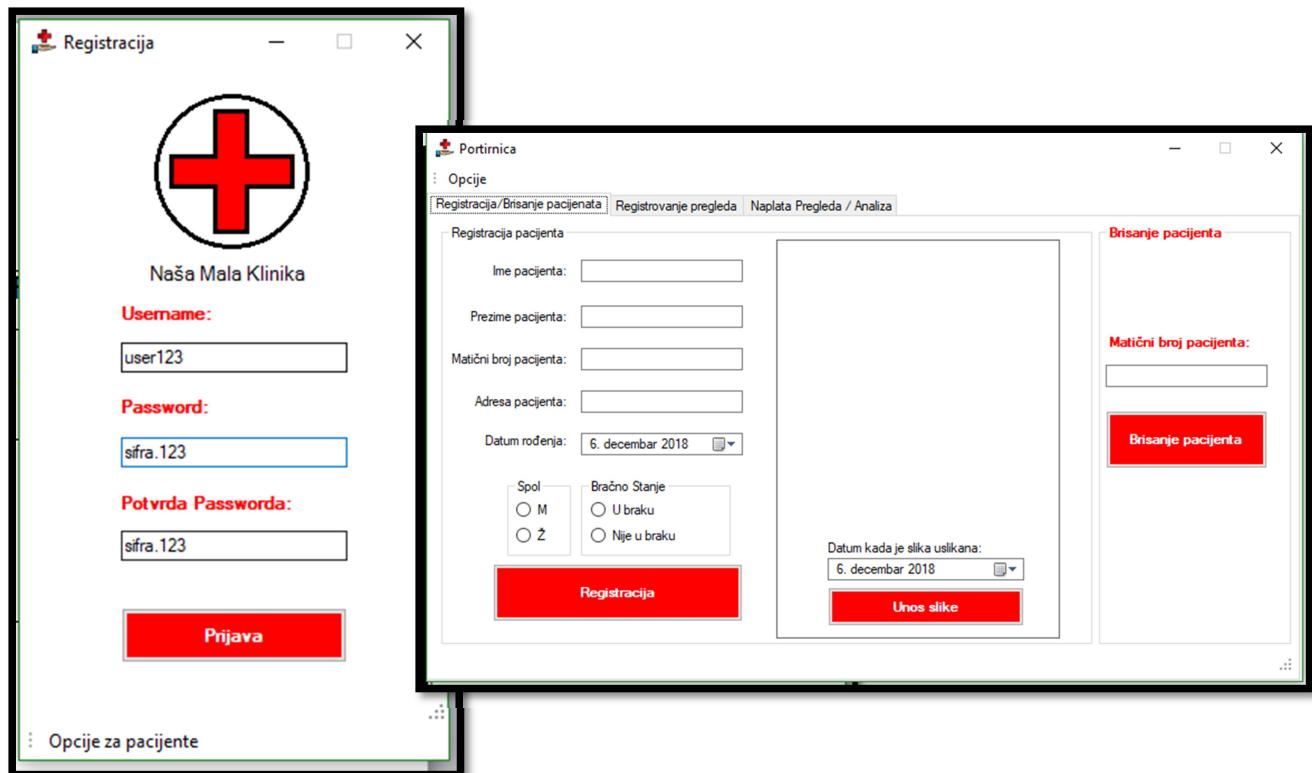
Varijabla	Vrijednost	Vrijednosti za validne EC	Granične vrijednosti	Nevalidni EC	Reprezentativne vrijednosti za nevalidne EC
Username	(1) Validan (2) Nevalidan	Username1, user123		Bilo šta što ne zadovoljava regex	-username, !mojaccount!
Password	(1) Validan (2) Nevalidan	sifra.123, novasifra.12345		Bilo šta što ne zadovoljava regex	Sifra, MOJASIFRA123
Potvrda passworda	(1) Validan (2) Nevalidan	sifra.123, novasifra.12345		Ako nije zadovoljena jednakost stringova	Sifra1, MOJASIFRA1234

Formirajmo sad slučajeve testiranja na osnovu naših EC klasa (uzeo sam 1 slučaj za validne EC i 3 slučaja za nevalidne EC, jer se ostali svedu na te, traži se svakako nekoliko slučajeva):

Tip	Broj	Username	Password	Cross validacija	Rezultat testnog slučaja
<b>za Validne EC</b>	1	user123	sifra.123	sifra.123	Ok, Portirnica otvorena
<b>za nevalidne EC</b>	2 3 4	u1 user123 user123	sifra.123 MOJASIFRA1 sifra.123	sifra.123 MOJASIFRA1 sifra.124	Nevalidan username Nevalidan password Nevalidna cross validacija

Testirajmo sada naše testne slučajeve.

Test broj 1(unos i rezultat respektivno):



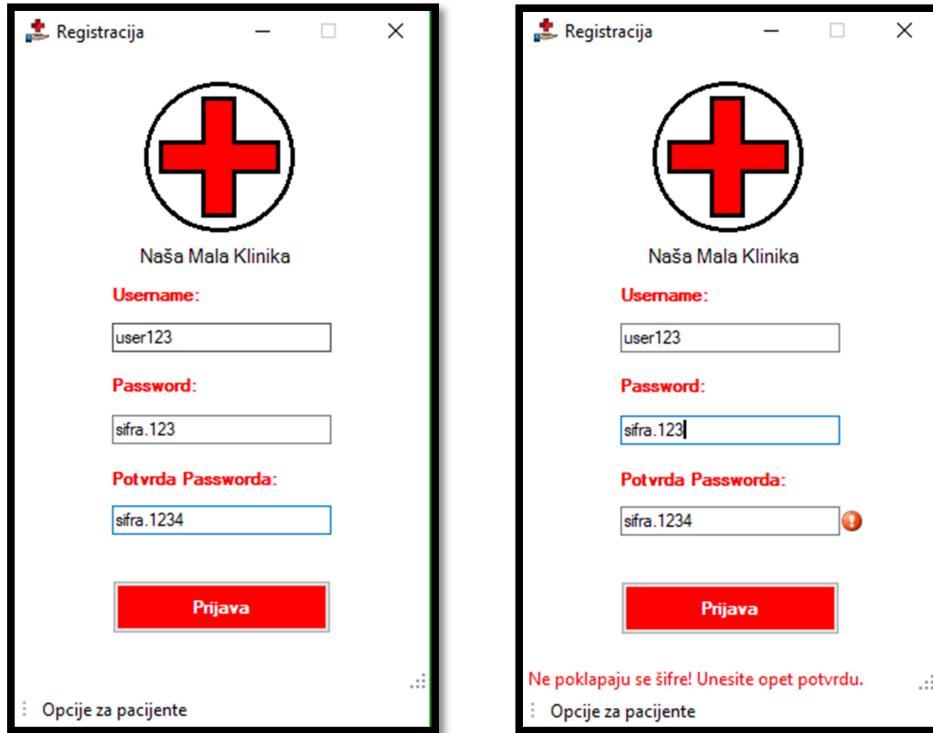
Test broj 2:

The application window has a red cross logo at the top left and the text 'Naša Mala Klinika' below it. It contains four input fields: 'Username:' with value 'u1', 'Password:' with value 'sifra.123', 'Potvrda Passworda:' with value 'sifra.123', and a 'Prijava' button. At the bottom, there is a link 'Opcije za pacijente'. The second screenshot shows the same interface, but the 'Username:' field now has a red exclamation mark icon, indicating an error.

Test broj 3:

The application window has a red cross logo at the top left and the text 'Naša Mala Klinika' below it. It contains four input fields: 'Username:' with value 'user123', 'Password:' with value 'MOJASIFRA1', 'Potvrda Passworda:' with value 'MOJASIFRA1', and a 'Prijava' button. At the bottom, there is a link 'Opcije za pacijente'. The second screenshot shows the same interface, but the 'Password:' field now has a red exclamation mark icon, indicating an error.

i na kraju test broj 4:



c) Demonstrirati *black box* testiranje - odabratи neku funkcionalnost aplikacije (ukoliko ne odgovara sajt dobijen u zadatku 1 da se prikaže ova tehnika možete odabratи npr. projekat sa OOAD-a, projekat ili zadaću nekog drugog predmeta, ili neki drugi web sajt) i na osnovu dokumentacije, *helpa* (ili izgleda ekrana) formirati tabele odluke i na osnovu toga testne slučajeve i scenarije. Izvršiti manuelno testiranje na osnovu testnih slučajeva (dovoljno za nekoliko testova) i sačiniti izvještaj o testiranju. ( 1 bod )

Odrabratи ћemo istu aplikaciju i funkcionalost kao pod b, samo što ћemo manuelno testiranje odraditi za slučajeve koje nismo gore.

Prvo ћemo formirati tabelu odluke.

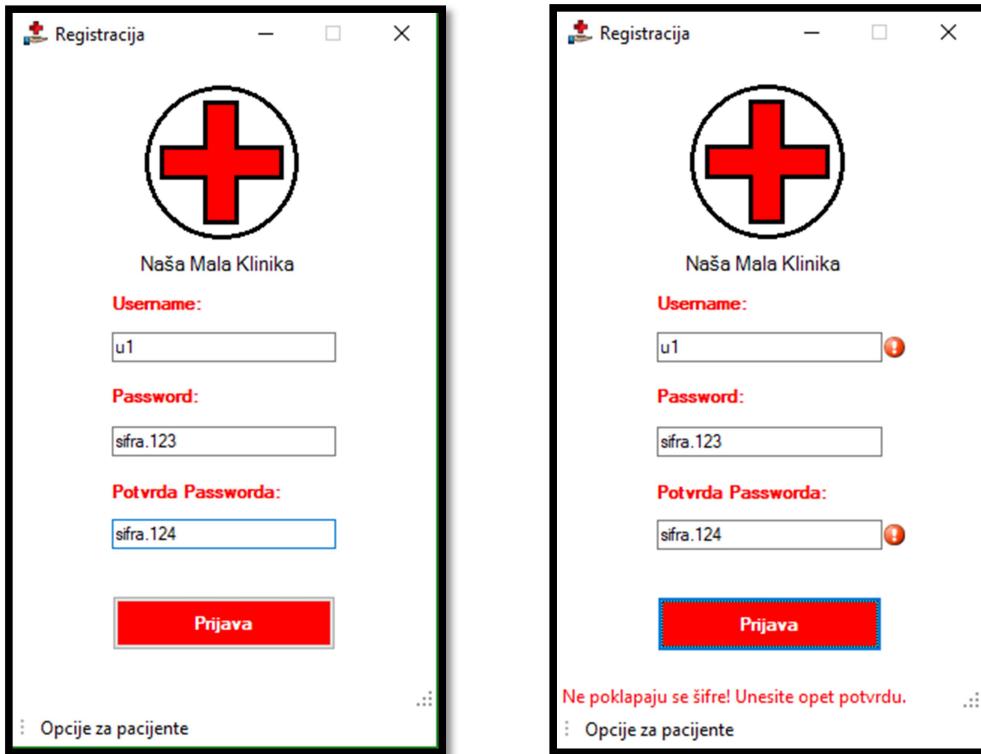
Uslov	Pravilo 1	Pravilo 2	Pravilo 3	Pravilo 4	Pravilo 5	Pravilo 6	Pravilo 7	Pravilo 8
Username	F	T	F	T	F	T	F	T
Password	F	F	T	T	F	F	T	T
Cross password	F	F	F	F	T	T	T	T
Izlaz (P – pogrešni pristupni podaci, PO – portirnica)	P	P	P	P	P	P	P	PO

Vidimo da smo dobili sve varijante ulaznih vrijednosti (tj. kada je validacija zadovoljena to znači da je T, odnosno F kada nije) u odnosu na formiranje po ekvivalentnim klasama. Testirati ćemo pravilo 1 i pravilo 3 jer takve varijante nismo testirali u zadatku pod b).

Pravilo 1 (testiranje kada oba regexa nisu zadovoljena i cross validacija):

Pravilo 3 (testiranje kada 2 uslovanisu zadovoljena od 3):

Napomena: Ostalo je u kodu da se uvijek ispisuje zadnja validacija koja nije zadovoljena u textu crvenom bojom.



### ZADATAK 3: Istraživanje 3 boda

**Objasniti namjenu i način provođenja uzrok-efekat black-box tehnike testiranja (Cause and Effect Graph Testing Technique).**

Uzrok-efekt back-box tehnika testiranja u biti predstavlja vezu između izlaza i svih stvari koje utiču na taj izlaz. Također u literaturi se spominje kao Ishikawa dijagram, jer po načinu izgleda kao fish-bone graf, koji je izmislio Ishikawa.

Namjena same metode jest da otkrije sve uzroke odnosno stvari koje utiču na sami izlaz, i na osnovu tih informacija da se mogu formirati testni slučajevi. Glavna prednost ove metode jeste što omogućava logičko povezivanje samih izlaza tj. efekata sa ulazima odnosno uzorcima tj. u testing maniru početnih vrijednosti za naše testne slučajeve.

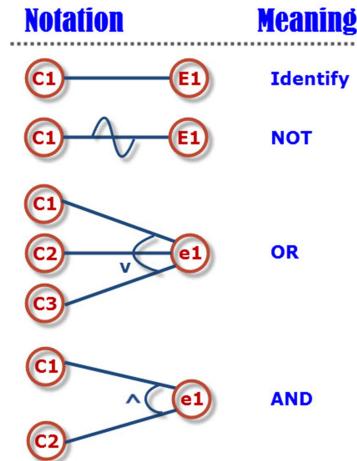
Zahvaljujući ovoj prednosti mogu se konstruisati razni testni slučajevi, tako da na osnovu ulaza i izlaza (pomoću ove metode) možemo izvršiti black box testiranje na osnovu dobivenih testnih slučajeva.

**Objasnimo sada način provođenja uzrok-efekat graf testing tehnike.**

Imamo 4 iteracije za kreiranje naših testnih slučajeva:

1. Prepoznati i opisati efekte i uzroke našeg problema
2. Napraviti uzrok-efekat graf tj. nacrtati na osnovu naših efekata i uzoraka.

Primjer crtanja za osnovne operacije:



3. Konvertovati naš uzrok-efekat graf u tabelu odluke
4. Iz tabele odluke pročitati testne slučajeve, svaka kolona odgovara jednom testnom slučaju

**Referencirati izvore koje ste koristili. Nije dozvoljeno direktno prevodenje i preuzimanje teksta.**

Reference:

1. <http://www.erosesengunthar.ac.in/dept/lm/CSE/ST/Cause%20&%20Effect%20Graph.ppt>
2. [https://www.tutorialspoint.com/software\\_testing\\_dictionary/cause\\_effect\\_graph.htm](https://www.tutorialspoint.com/software_testing_dictionary/cause_effect_graph.htm)
3. <https://www.softwaretestingclass.com/what-is-cause-and-effect-graph-testing-technique/>

**Ilustrovati način primjene ove tehnike na svom originalnom primjeru.**

Na predmetu OR profesor je zadao sljedeći zadatak. Potrebno je pročitati skraćenice za određene države, te skraćenice sadrže po 2 karaktera. Međutim nas zanimaju samo države koje imaju prvi karakter ASCII vrijednosti ili 68 ili 69. Drugi karakter koji se pročita mora biti malo slovo jer inače ne bi bila oznaka za državu. Ukoliko je tražena država zadovoljena treba se ispisati „Traženi država je zadovoljena“.

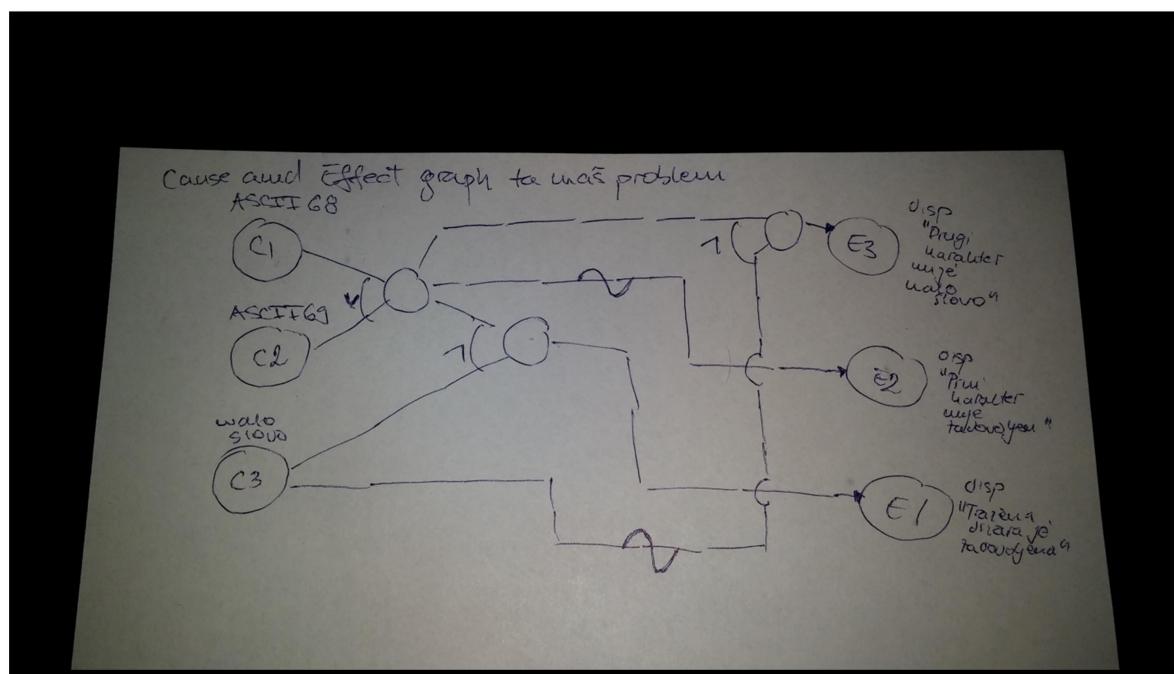
Ukoliko prvi karakter nije zadovoljen treba ispisati „Prvi karakter nije zadovoljen“. Ukoliko drugi karakter nije malo slovo treba ispisati „Drugi karakter nije malo slovo“. Prioritet pri ispisu (ukoliko oba slučaja nisu tačna) ima prvi karakter.

#### **Prateći gornja 4 koraka imamo sljedeće rješenje:**

1. Imamo 3 moguća efekta naše postavke, prvi je da ukoliko su oba uslova zadovljena da se ispiše „Tražena država je zadovoljena“ ovo je naše E1, zatim ukoliko prvi karakter nije zadovoljen (nevezano jesli drugi zadovoljen ili ne) da se ispiše „Prvi karakter nije zadovoljen“ ovo je naše E2, zatim ukoliko je prvi karakter zadovoljen a nije drugi da se ispiše „Drugi karakter nije malo slovo“ ovo je naše E3.

Imamo također 3 uzroka, prvi uzrok je unos karaktera ASCII vrijednosti 68 što ćemo označiti kao C1, zatim drugi sa 69 što ćemo označiti kao C2, i treći unos malog slova što ćemo označiti kao C3.

2. Nacrtajmo naš uzrok-efekat graf.



3. Konvertujmo ovaj graf u tabelu odluke:

Ulazni parametri/uslovi	Pravilo 1	Pravilo 2	Pravilo 3	Pravilo 4	Pravilo 5	Pravilo 6	Pravilo 7	Pravilo 8
C1	0	1	0	1	0	1	0	1
C2	0	0	1	1	0	0	1	1
C3	0	0	0	0	1	1	1	1
Izlazne vrijednosti	E2	E3	E3	E3	E2	E1	E1	E1

4. Imamo 8 pravila odnosno 8 testnih slučajeva koje možemo napisati za naš problem. 0 znači nevalidna vrijednost za ulazne varijable, dok 1 znači validna vrijednost za ulazne varijable, pri čemu svaka od vrijednosti ulaza C1,C2,C3 je opisana u prethodnom dijelu zajedno sa izlaznim funkcionalnostima E1,E2,E3.