

TESTIRANJE SOFTVERA - VEŽBE 02

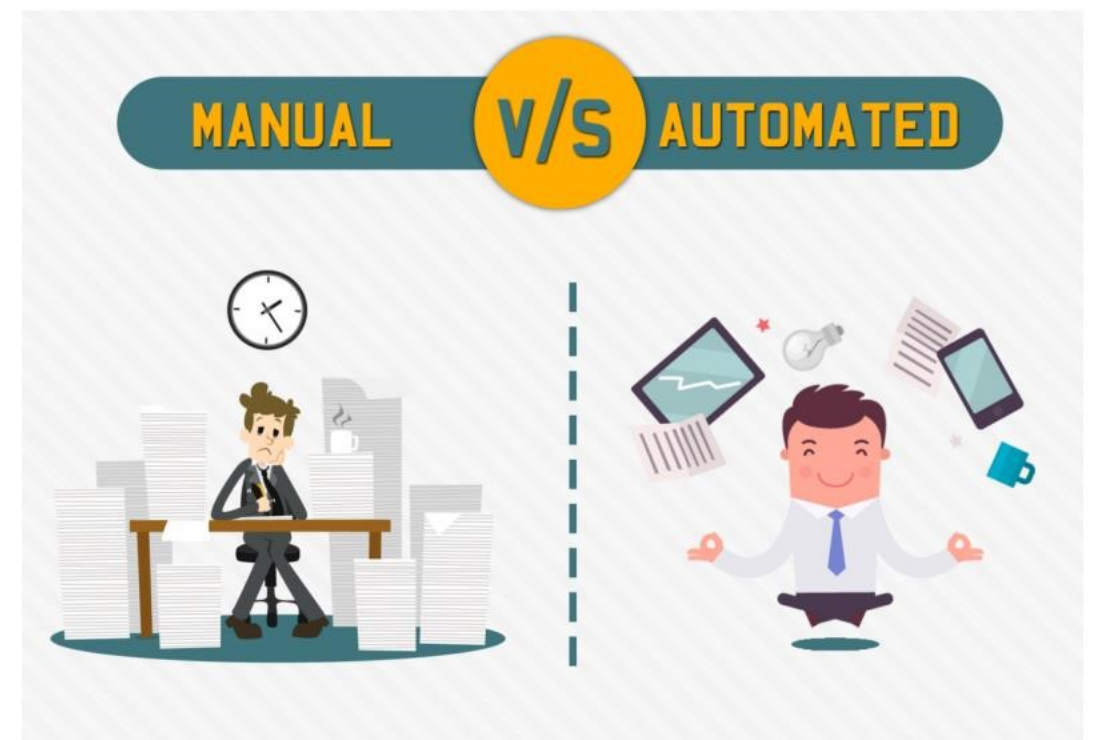
---

# MANUELNO TESTIRANJE

---

# TESTIRANJE SOFTVERA

- ▶ Proces kojim se utvrđuje da softver:
  - ▶ radi bez grešaka
  - ▶ efikasno ispunjava zahteve korisnika
  - ▶ uz obradu izuzetnih i graničnih slučajeva
- ▶ Testiranje softvera se u opštem slučaju može klasifikovati na:
  - ▶ Manuelno
  - ▶ Automatsko



---

# MANUELNO TESTIRANJA

- ▶ Manuelno testiranje je proces u kojem QA inženjeri izvršavaju testove jedan po jedan na individualan način
- ▶ Svrha manualnog testiranja je da se otkriju greške i problemi u softveru pre nego što isti ode u produkciju
- ▶ Tom prilikom QA inženjeri validiraju osnovne funkcionalnosti softvera, najčešće u odnosu na datu specifikaciju
- ▶ Analitičar izvršava test slučajeve i kreira sumarni izveštaj o greškama manualno, ne koristeći dodatne softverske alate
- ▶ Tester koristi planove testiranja, test slučajeve i scenarije testiranja u cilju testiranja softver

---

# AUTOMATSKO TESTIRANJE

- ▶ U automatskom testiranju softvera, tester pišu kod/testne skripte da bi automatizovali izvršenje testa, tom prilikom koriste alate za automatizaciju
- ▶ Cilj je skratiti vreme trajanja testiranja
- ▶ Automatsko testiranje se u potpunosti oslanja na unapred pripremljeni test koji se pokreće automatski sa ciljem da uporedi stvarni rezultat sa očekivanim rezultatima
- ▶ Automatsko testiranje omogućava da reperativno izvršavanje testova bez intervencije manuelnog testera. Iako se svi procesi izvode automatski, automatizacija zahteva izvestan manuelni napor u cilju kreiranja inicijalne skripte za testiranje

---

# MANUELNO VS AUTOMATSKO TESTIRANJE

Aspekt testiranja	Manuelno	Automatski
Izvršavanje	QA tester	Automatski korišćenjem skripti i dodatnih softverskih alata
Efikasnost	Vremenski zahtevno i manje efikasno	Veći broj izvršeni testova u istom vremenskom roku, povećana efikasnost
Tipovi zadataka	Svi zadaci su manualni	Većina zadataka je automatizovana, uključujući i simulacije realnog korisnika
Pokrivenost testovima	Teško je obezbediti i meriti pokrivenost koda testovima	Pokrivenost se jako lako prati i obezbeđuje
Programersko znanje	Nije nužno	Neophodno
Pouzdanost	Nije preterano pouzdano, oslanjeno na čoveka koji može da pravi greške	Veoma pouzdano, s ozbirom da se oslanjamo na skripte
Promene dizajna	Ne utiču na testove	Čak i trivijalne izmene UI dovode do promena u testovima

---

# MANUALNO TESTIRANJE – PRO AND CONS

## ► Prednosti:

1. Brza i tačna vizuelnu potvrda rada
2. Jeftinije je jer se ne trošiti budžet na alate i procese za automatizaciju
3. Faktor ljudskog rasuđivanje i intuicije

## ► Mane:

1. Manje pouzdan metod testiranja jer ga sprovodi čovek (povećana sklonost ka greškama)
2. Proces manuelnog testiranja se ne može snimiti, tako da nije moguće ponovo koristiti isti test.
3. U ovoj metodi testiranja, određene zadatke je teško izvršiti ručno, što može zahtevati dodatno vreme faze testiranja softvera

---

# AUTOMATSKO TESTIRANJE – PRO AND CONS

## ► Prednosti:

1. Automatsko testiranje omogućava pronalaženje većeg broja grešaka u poređenju sa ljudskim testerom
2. Pošto je veći deo procesa testiranja automatizovan, predstavlja brz i efikasan proces
3. Proces automatizacije se može snimiti. Ovo omogućava ponovno korišćenje i izvršavanje iste vrste testiranja
4. Automatsko testiranje se sprovodi pomoću softverskih alata, tako da radi bez zamora za razliku od ljudi u manuelnom testiranju
5. Lako može povećati produktivnost jer pruža brz i precizan rezultat testiranja
6. Automatsko testiranje podržava različite aplikacije

## ► Mane:

1. Bez ljudskog faktora, teško je dobiti uvid u vizuelne aspekte korisničkog interfejsa (boje, font, veličine, kontrast ili veličine dugmadi)
2. Alati za pokretanje automatizovanog testiranja mogu biti skupi, što može povećati cenu projekta testiranja
3. Svaki alat za automatizaciju ima svoja ograničenja koja smanjuju obim automatizacije
4. Otklanjanje grešaka u test skripti je još jedan veliki problem u automatizovanom testiranju. Održavanje testa je skupo

---

# POZITIVNO TESTIRANJE

Enter Only Numbers

99999

**Positive Testing**

- ▶ Positive Testing je vrsta testiranja koja se izvodi na softverskoj aplikaciji obezbeđivanjem validnih skupova podataka kao ulaza
- ▶ Proverava da li se softverska aplikacija ponaša kako se očekuje sa pozitivnim ulazima ili ne. Pozitivno testiranje se vrši kako bi se proverilo da li softverska aplikacija radi upravo ono što se od nje očekuje
- ▶ Primer:
  - ▶ U aplikaciji postoji okvir za tekst koji može da prihvati samo brojeve. Unošenje vrednosti do 99999 biće prihvatljivo za sistem i bilo koje druge vrednosti osim ove ne bi trebalo da budu prihvatljive. Pozitivno testiranje podrazumeva unošenje važećih ulaznih vrednosti od 0 do 99999, kao i proveru prihvaćenosti vrednosti od strane sistema



---

# NEGATIVNO TESTIRANJE

Enter Only Numbers

abcdef

**Negative Testing**

- ▶ Negativno testiranje je metoda testiranja koja se izvodi na softverskoj aplikaciji obezbeđivanjem nevažećih ili neodgovarajućih skupova podataka kao ulaza
- ▶ Proverava da li se softverska aplikacija ponaša kako se očekuje sa negativnim ili neželjenim korisničkim unosima. Svrha negativnog testiranja je da osigura da se softverska aplikacija ne ruši i da ostane stabilna sa nevažećim unosima podataka
- ▶ Primer:
  - ▶ Negativno testiranje se može izvršiti unošenjem znakova od A do Z ili od a do z. Ili softverski sistem ne bi trebalo da prihvati vrednosti ili bi trebalo da izbací poruku o grešci za ove nevažeće unose podataka

# TEHNIKE ZA ODREĐIVANJE SKUPOVA PODATAKA

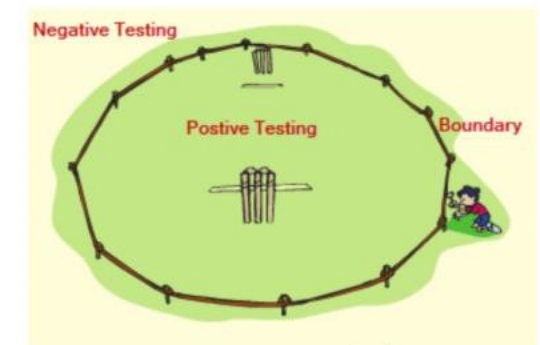
▶ Sledeće tehnike se koriste za pozitivnu i negativnu validaciju testiranja:

- ▶ Analiza graničnih vrednosti
- ▶ Ekvivalentno particionisanje

▶ Analiza graničnih vrednosti

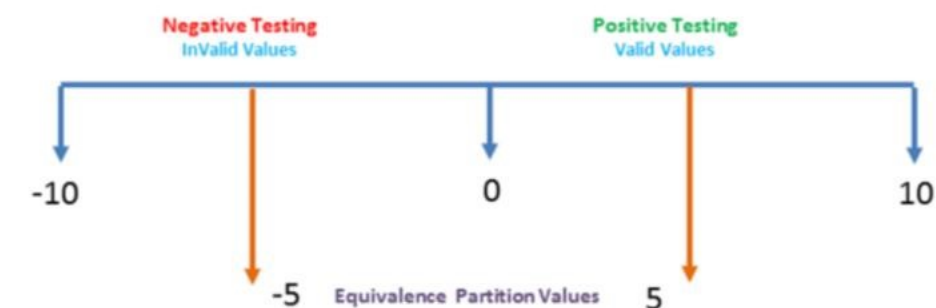
- ▶ Ovo je jedna od tehnika testiranja softvera u kojoj su testni slučajevi dizajnirani da uključuju vrednosti na granici. Ako se ulazni podaci koriste unutar granica graničnih vrednosti, onda se kaže da je to pozitivno testiranje. Ako su ulazni podaci izabrani izvan granica graničnih vrednosti, onda se kaže da je to negativno testiranje

- ▶ Primer: Sistem može da prihvati brojeve od 0 do 10 numeričkih vrednosti. Svi ostali brojevi su nevažeće vrednosti. U okviru ove tehnike, granične vrednosti -1,0,1 i 9,10,11 će biti testirane



▶ Ekvivalentno particionisanje

- ▶ Ovo je tehnika testiranja softvera koja deli ulazne podatke na mnogo particija. Vrednosti iz svake particije moraju biti testirane najmanje jednom. Particije sa važećim vrednostima se koriste za pozitivno testiranje. Dok se particije sa nevažećim vrednostima koriste za negativno testiranje
- ▶ Primer: Numeričke vrednosti od 0 do 10 mogu podeliti na dve (ili tri) particije. U našem slučaju, imamo dve particije -10 do -1 i 0 do 10. Vrednosti uzorka (5 i -5) se mogu uzeti iz svakog dela da bi se testirali scenariji



---

# TEST SLUČAJ

- ▶ Test slučaj (*eng. test case*) je skup uslova ili promenljivih pod kojima tester utvrđuje da li softver zadovoljava zahteve (*eng. requirements*) i funkcioniše ispravno
- ▶ Test slučaj se može zamisliti i kao skup uputstava korak po korak za proveru da li se softver ili neka njegova komponenta ponaša onako kako je definisano specifikacijom
- ▶ Saveti za pisanje dobrih test slučajeva:
  - ▶ Test slučajevi treba da budu transparentni i jasni
  - ▶ Test slučajeve kreirati tako da na umu krajnji korisnik
  - ▶ Izbegavati ponavljanje testnih slučajeva
  - ▶ Nikada ne pretpostavljati funkcionalnosti i karakteristike softverske aplikacije tokom pripreme test slučaja
  - ▶ Test slučajevi moraju biti lako prepoznatljivi

---

# TEST SLUČAJ

- ▶ Test slučaj obično sadrži:
  - ▶ Jedinstveni identifikator
  - ▶ Naslov
  - ▶ Test podaci - Dataset koji je korišćen za testiranje
  - ▶ Preduslove - Koraci koje se moraju uraditi pre nego što se počnu izvršavati koraci test slučaja. To može biti instalacija nekog dodatnog softvera ili dovođenje aplikacije u određeno stanje...
  - ▶ Korake testiranja
    - ▶ Svaki test slučaj mora imati korake koje je potrebno sprovesti. Ovo je neophodno zapisati (dokumentovati), u slučaju da tester koji je radio na test slučaju napusti firmu ili ode na odmor ili uzme bolovanje. Ovi koraci izgledaju mnogo jednostavno, ali ukoliko je u pitanju komplikovanja funkcionalnost tada i broj koraka može postati veći
  - ▶ Očekivani rezultat - Izlaz koji se očekuje od sistema
  - ▶ Stvarni rezultat - Izlaz koji smo zapravo dobili od sistema
  - ▶ Posledice - Sve što se dogodilo kao posledica izvršenog test slučaja
  - ▶ Pass/fail - Ako se stvarni rezultati poklapaju sa očekivanim, test je prošao. Ako nije, test je pao

---

# TEST SCENARIO

- ▶ Scenario testiranja je detaljan dokument koji sadrži niz test slučajeva koji pokrivaju end-to-end funkcionalnost softverske aplikacije
- ▶ Testni scenario je klasifikacija zahteva koji se mogu testirati na visokom nivou. Ovi zahtevi su grupisani na osnovu funkcionalnosti modula i dobijeni iz slučajeva korišćenja
- ▶ U scenariju testiranja, testeri treba da se postave na mesto korisnika jer testiraju softversku aplikaciju sa tačke gledišta korisnika. Priprema scenarija je najkritičniji deo i potrebno je potražiti savet ili pomoć od kupaca, zainteresovanih strana ili programera da bi se pripremio scenario
- ▶ Saveti za pisanje dobrih test scenarija:
  - ▶ Svaki test scenario treba da bude vezan za najmanje jedan zahtev ili korisničku priču prema metodologiji projekta
  - ▶ Pre kreiranja testnog scenarija koji verifikuje više zahteva odjednom, uverite se da postoji testni scenario koji proverava taj zahtev izolovano
  - ▶ Izbegavati kreiranje preterano komplikovanih testnih scenarija koji obuhvataju više zahteva
  - ▶ Broj scenarija može biti veliki, a sve ih je skupo pokrenuti. Na osnovu prioriteta kupaca pokrenite samo odabrane testne scenarije

---

# TEST SCENARIO VS TEST SLUČAJ

## Test Scenario

Testni scenario je koncept

Test Scenario je funkcionalnost visokog nivoa

Test scenariji su izvedeni iz zahteva/korisničkih priča

Test scenario odgovara na pitanje „Koja funkcionalnost treba da se testira“

Testni scenariji imaju više test slučajeva

Scenariji pojedinačnih testova se nikada ne mogu ponoviti

Potrebna kratka dokumentacija

Neophodne su brainstorming sesije da bi se završio test scenario

Ušteda vremena jer sitni detalji nisu potrebni

Troškovi održavanja su niski jer su potrebni resursi niski

## Test Case

Testni slučajevi su konkretne situacije kojima se verifikuje koncept

Testni slučajevi su detaljna procedura za testiranje funkcionalnosti visokog nivoa

Testni slučajevi su izvedeni iz testnih scenarija

Testni slučajevi odgovaraju na pitanje „Kako testirati funkcionalnost“

Testni slučaj može, ali ne mora biti povezan sa više testnih scenarija

Jedan test slučaj se može koristiti više puta u različitim scenarijima

Potrebna je detaljna dokumentacija

Potrebno je detaljno tehničko poznavanje softverske aplikacije

Zahteva mnogo vremena jer treba voditi računa o svakom sitnom detalju

Troškovi održavanja su visoki jer su potrebni resursi visoki

---

# DOKUMENTOVANJE

- ▶ Na kraju je potrebno negde i zabeležiti rezultate koje smo dobili za različite test slučajeve
- ▶ Postoji template koji manuelni testeri koriste za test slučajeve
- ▶ Može se razlikovati u zavisnosti od kompanije do kompanije
- ▶ Najčešće je to word ili excel dokument
- ▶ Mogu se naći različiti template-ti za test slučajeve: <https://www.softwaretestinghelp.com/test-case-template-examples/>
- ▶ Takođe, danas na tržištu postoji i mnoštvo alata koji pomažu pri dokumentovanju test slučajeva

---

# PRIMER

- ▶ Test scenario
  - ▶ Proveriti funkcionalnost prijave na sistem
- ▶ Test cases
  - ▶ Unos validnog korisničkog imena i lozinke
  - ▶ Unos nevalidnog korisničkog imena i lozinke
  - ▶ Ostaviti prazno polje za korisničko ime
  - ▶ ...



# PRIMER

Test Scenario ID		Login-1		Test Case ID		Login-1A	
Test Case Description		Login – Valid username and password		Test Priority		High	
Pre-Requisite		A valid user account		Post-Requisite		NA	
Test Execution Steps:							
S.No	Action	Inputs	Expected Output	Actual Output	Test Browser	Test Result	Test Comments
1	Launch application	https://someapp.com/	Some App home page	Some App home page	GC-105.98	Pass	[Marc 10/10/2022 11:44 AM]: Launch successful
2	Open login form		Login form opened	Login form opened	GC-105.98	Pass	[Marc 10/10/2022 11:45 AM]: Form opened
2	Enter correct Email & Password and hit login button	Email id : test@xyz.com Password: *****	Login success	Login success	GC-105.98	Pass	[Marc 10/10/2022 11:45 AM]: Login successful

---

# TESTIRANJE WEB APLIKACIJE

- ▶ Pretpostavimo da razvijamo aplikacija podržava sledeće funkcionalnosti:
  - ▶ Forme sa raznim poljima
  - ▶ Aplikacija je u interakciji sa bazom podataka
  - ▶ Različiti kriterijumi filtera za pretragu i prikaz rezultata
  - ▶ Upload slike/fajla
  - ▶ Funkcionalnost slanje e-pošte
  - ▶ Export podataka
- ▶ Za svaku od ovih funkcionalnosti postoje grupe test scenarija i slučajeva koje su podrazumevane u fazi testiranja
- ▶ Šta biste vi testirali?

---

# TESTIRANJE FORME

1. Sva obavezna polja treba da budu proverena i označena zvezdicom (\*)
2. Poruke o grešci pri validaciji polja treba da budu prikazane ispravno
3. Sve poruke o greškama treba da budu prikazane u istom CSS stilu (npr., koristeći crvenu boju)
4. Tooltips tekst treba da ima smisla
5. Drop-down polja treba da imaju prvu stavku kao praznu ili tekst kao što je "Izaberi"
6. Select/deselect opcija treba da bude obezbeđena ako stranica podržava funkcionalnost dodavanja/brisanja/ažuriranja nekih stavki
7. Vrednosti iznosa treba da budu prikazane sa ispravnim simbolima valute
8. Reset dugme treba da postavi podrazumevane vrednosti za sva polja
9. Polja za unos treba proveriti za maksimalnu vrednost polja. Ulazne vrednosti veće od navedenog maksimalnog ograničenja ne bi trebalo da budu prihvaćene ili uskladištene u bazi podataka
10. Proveriti koja polja podržavaju specijalne znake
11. Proveriti polja koja primaju samo numeričke vrednosti šta se desi kada se unese karakter koji nije broj. Trebalo bi da se pojavi odgovarajuća poruka
12. Proveriti unos negativnog broja ako je dozvoljen da se unese
13. Proveriti unos decimalnih brojeva
14. Korisnik ne bi trebalo da ima mogućnost da dva puta pošalje istu popunjenu formu tako što klikne 2 puta na Submit dugme

---

# INTERAKCIJA SA BAZOM PODATAKA

1. Proveriti da li se ispravni podaci čuvaju u bazi podataka nakon uspešnog prosleđivanja
2. Proveriti vrednosti za kolone koje ne prihvataju vrednosti "null"
3. Tabele treba da imaju kolonu za ključ
4. Za svako dodavanje ili izmenu nekog entiteta trebalo bi da postoje logovi
5. Treba kreirati potrebne indekse tabele
6. Proveriti da li se radi commit samo kada je operacija uspešno završena
7. Proveriti da li se radi roll back u slučaju neuspešnih transakcija
8. Proveriti da li ulazni podaci nisu skraćeni prilikom čuvanja. Dužina polja prikazana korisniku na stranici i u šemi baze podataka bi trebalo da bude ista
9. Proveriti numerička polja sa minimalnim, maksimalnim i decimalnim vrednostima
10. Proveriti numerička polja sa negativnim vrednostima
11. Proveriti da li su opcije radio dugmeta i drop-down (padajuće) liste ispravno sačuvane u bazu podataka
12. Proveriti da li su polja baze podataka dizajnirana sa ispravnim tipom podataka i dužinom podataka
13. Proveriti da li su sva ograničenja tabele kao što su Primarni ključ, Strani ključ itd. dobro implementirana
14. Razmake na početku i kraju polja za unos treba odseći (trim) pre nego što podatke sačuvate u bazi podataka
15. "Null" vrednosti ne bi trebalo da budu dozvoljene za kolonu primarnog ključa

---

# RAZLIČITI KRITERIJUMI FILTERA ZA PRETRAGU I PRIKAZ REZULTAT

1. Korisnik bi trebalo da bude u mogućnosti da filtrira rezultate koristeći sve parametre na stranici
2. Funkcionalnost pretraživanja treba da učitava stranicu sa rezultatima koji zadovoljavaju sve parametre pretrage koje je korisnik izabrao
3. Kada postoji najmanje jedan kriterijum filtera potreban za izvršavanje operacije pretraživanja, proverite da je odgovarajuća poruka o grešci prikazana kada korisnik prosledi stranicu bez izabranog kriterijuma filtera
4. Kada izbor kriterijuma filtera nije obavezan, korisnik bi trebalo da bude u mogućnosti da prosledi stranicu, a podrazumevani kriterijumi pretrage treba da se koriste za rezultate upita
5. Treba prikazati odgovarajuće poruke za proveru valjanosti za sve nevažeće vrednosti kriterijuma filtera

---

# UPLOAD SLIKE/FAJLA

1. Proverite da li je putanja slike/fajla validn
2. Proverite dodavanje (upload) i izmenu dodate slike
3. Proveriti funkcionalnost dodavanja (upload) slike pomoću za različite ekstenzije (npr., JPEG, PNG, BMP itd.)
4. Proveriti funkcionalnost dodavanja (upload) slike/fajla sa slikama koje imaju space ili bilo koji drugi dozvoljeni specijalni znak u putanji datoteke
5. Proveriti otpremanje (upload) slike/fajla sa veličinom slike većom od maksimalne dozvoljene veličine. Trebalo bi prikazati odgovarajuće poruke o greškama
6. Proveriti funkcionalnost otpremanja slike sa tipovima datoteka osim slika (Na primer, txt, doc, pdf, exe itd.). Trebalo bi prikazati odgovarajuću poruku o grešci
7. Proverite da li su slike navedene visine i širine (ako su definisane)
8. Progres bar bi trebalo da se pojavi prilikom upload-a slike velike veličine
9. Proveriti da li Cancel dugme radi u toku procesa upload-a
10. Proveriti funkcionalnost upload-a više slika
11. Proveriti kvalitet upload-ovane slike. Kvalitet ne bi trebalo da se promeni nakon procesa upload-
12. Proveriti da li je korisnik u mogućnosti da koristi/pregleda upload-ovane slike

---

# FUNKCIONALNOST SLANJA E-POŠTE

1. E-mail adrese treba proveriti pre slanja
2. Specijalne znake u sadržaju mejla treba pravilno prikazati
3. Specifične karaktere nekog jezika (npr. znakovi ruskog, kineskog ili nemačkog jezika) bi trebalo pravilno prikazati
4. Naslov mejla ne bi trebalo da bude prazan
5. Ime pošiljaoca e-pošte ne bi trebalo da bude prazno
6. Slanje e-pošte treba proveriti preko različitih klijenta e-pošte kao što su Outlook, Gmail, Hotmail, Yahoo! pošta itd
7. Proveriti funkcionalnost slanja e-pošte koristeći polja TO, CC i BCC.
8. Proveriti header i footer e-pošte za logotip kompanije, smernice privatnosti i druge veze
9. Proveriti mejlove sa priložima

---

# FUNKCIONALNOST EKSPORTA PODATAKA

1. File u koji se vrši export podataka bi trebalo da ima odgovarajuću ekstenziju
2. Naziv esportovane datoteke bi trebalo da bude po standardima, npr. ako ime datoteke koristi vremensku oznaku, trebalo bi da bude pravilno zamenjeno stvarnim vremenom u trenutku exporta fajla
3. Proveriti da li postoji format datuma ako exportovana Excel datoteka sadrži kolone datuma
4. Proveriti formatiranje brojeva. Formatiranje bi trebalo da bude isto kao što je prikazano na stranici
5. Exportovana datoteka treba da ima kolone sa odgovarajućim imenima kolona
6. Podrazumevano sortiranje stranica treba da se sprovede i u exportovanoj datoteci
7. Proveriti da li su podaci koji su prikazani na stranici i exportovani u Excel fajl isti
8. Proveriti funkcionalnost exporta kada je paginacija omogućena
9. Proveriti da li dugme za export prikazuje odgovarajuću ikonu u skladu sa ekstenzijom exportovane datoteke, na primer, ikona Excel datoteke za xls datoteke
10. Proveriti funkcionalnost exporta datoteka sa veoma velikom veličinom
11. Proveriti funkcionalnost exporta za stranice koje sadrže specijalne znakove
12. Proveriti da li su ovi specijalni znakovi ispravno exportovani u Excel datoteku



# PRONADITE NEDOSTATKE

**New Registratio**

Choose User Id  Enter User ID

Password  Enter Password

Confirm Password

Name

Email  (Requires verification. Will not be published.)

Country

 Please enter the verification number exactly as shown in left.

---

# REŠENJE

- ▶ Polje User ID ne bi trebalo da prihvata specijalne karaktere
- ▶ Smernice za izbor lozinke bi trebalo da budu vidljive kao npr. "Lozinka mora sadržati brojeve, velika, mala slova i da ima min 8 karaktera", kao i pokazatelj koliko je jaka lozinka
- ▶ Polje za potvrdu lozinke (Confirm Password) ne prikazuje lozinku u šifrovanom modu
- ▶ Captcha nije čitljiva
- ▶ Ne postoji način da korisnik ponovo učitava Captcha-u
- ▶ Veličina polja i labela bi trebalo da bude ista za sva polja u formi
- ▶ Select polje za izbor države podrazumevano bi trebalo da pokazuje "Izaberi" ili "Select" umesto da ima neku državu kao podrazumevanu vrednost
- ▶ Dugme za registraciju bi trebalo da bude na dnu, a ne sa strane
- ▶ Register bi moglo da počinje sa velikim "R" a ne malim
- ▶ Ne postoji Cancel dugme ako korisnik hoće da otkaže proces registracije
- ▶ Ne postoji Close dugme ako korisnik hoće da zatvori stranu

---

# PRONAĐITE NEDOSTATKE

## Sign In

What is your e-mail address?

My e-mail address is

Do you have password?

☐ No, I am a new customer.

☒ Yes, I have a password:

[Forgotten your password?](#)

Sign in using our secure server

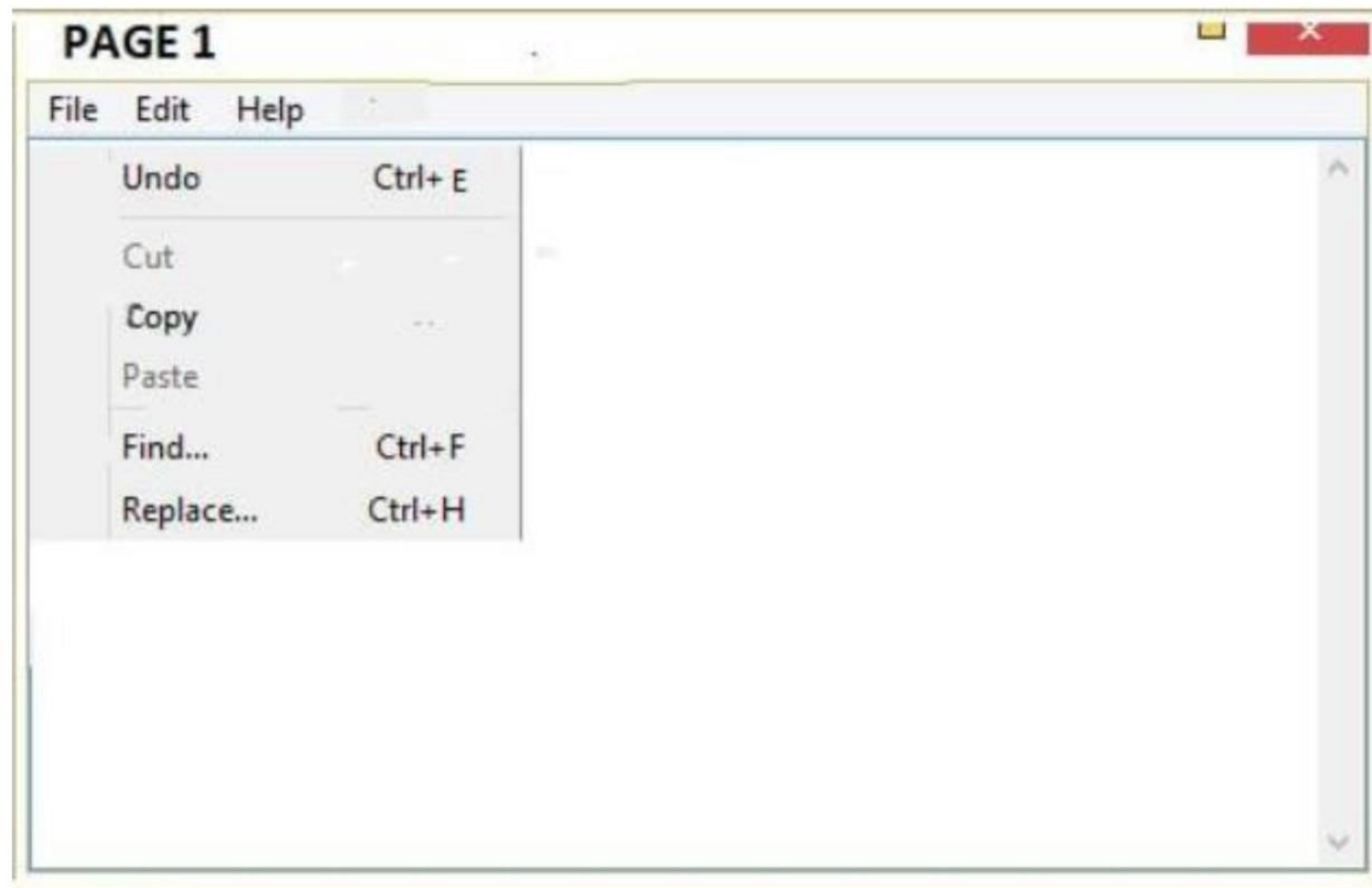


---

# REŠENJE

- ▶ Umesto pitanja "What is your e-mail address?" moglo bi da stoji samo "e-mail address"
- ▶  Prijava bi mogla da bude dostupna i preko user ID-a (korisničkog imena), a ne samo preko e-mail adrese
- ▶ Umesto da pita korisnika da selektuje da li je novi korisnik, sistem bi mogao na osnovu e-mail adrese da proveriti status korisnika
- ▶ Najčešće korisnik ne brine o tome da i sistem koristi siguran server, pretpostavlja se da je to podrazumevano. Dakle, dugme bi trebalo da bude jednostavno samo "Sign in" ili "Prijava"
- ▶ Link za pomoć bi trebalo da bude dostupan korisniku (Help)
- ▶ Ne bi bilo loše da postoji close ili cancel dugme

# PRONAĐITE NEDOSTATKE



---

# REŠENJE

- ▶ Ne postoji dugme za minimizovanje aplikacije
- ▶ Za Undo opciju najčešća prečica na tastaturi je Ctrl+Z i teško je korisnicima da se naviknu na drugačije prečice (Ctrl+E) za podrazumevanu operaciju kao što je Undo
- ▶ Za Cut, Copy i Paste nisu prikazane/obezbeđene prečice
- ▶ U naslovu se ne nalazi logo aplikacije