

**FFD**



**ResiMe**

**Plan testiranja**

# Sadržaj

1. Ciljevi	3
2. Opseg	3
3. Reference	3
4. Zahtevi za testiranje	4
5. Strategije testiranja	6
5.1 Tipovi testiranja	6
5.1.1 Testiranje podataka i integriteta baze podatak	6
5.1.2 Testiranje sistema	6
5.1.3 Testiranje poslovnog ciklusa	7
5.1.4 Testiranje korisnickog interfejsa	8
5.1.5 Testiranje performansi	8
5.1.6 Testiranje opterecenja	9
5.1.7 Testiranje kriticnih slucajeva	9
5.1.8 Testiranje volumena	10
5.1.9 Testiranje sigurnosti i kontrole pristupa	11
5.1.10 Testiranje otkaza i oporavka	11
5.1.11 Testiranje konfiguracije	12
5.1.12 Testiranje instalacije	12
5.2 Alati	13
6. Resursi	13
6.1 Radnici	13
6.2 Sistem	14
7. Glavne postavke testiranja	14
8. Rezultati testiranja	15
8.1 Model testiranja	15
9. Projektni zadaci	15

# Plan testiranja

## 1. Ciljevi

Ovaj document opisuje plan testiranja sistema ResiMe. Ciljevi dokumenta su:

- Identifikovati komponente sistema koje je potrebno testirati na osnovu postojećih informacija o projektu.
- Definirati zahteve testiranja na osnovu specifikacija i projektnih predloga viseg nivoa.
- Predložiti i detaljno opisati strategije testiranja koje će se koristiti tokom procesa provere sistema.
- Odrediti potrebne resurse, vreme i trud neophodne za sprovođenje testiranja.
- Navesti dokumente koji će sadržati rezultate i nalaze testiranja.

## 2. Opseg

Plan testiranja se odnosi na integraciju sistema i proveru izvršne verzije aplikacije.

Plan testiranja se odnosi na:

- Testiranje interfejsa prema različitim tipovima korisnika
- Testiranje komunikacije sa bazom podataka
- Testiranje komunikacije komponenti aplikacije
- Testiranje funkcionalnosti izvršne verzije aplikacije
- Testiranje kritičnih performansi aplikacije

## 3. Reference

Reference na osnovu kojih će biti napisan plan testiranja:

1. Vizija sistema
2. Specifikacija zahteva sistema
3. Detaljno – arhitekturni projekat

#### 4. Zahtevi za testiranje

U ovom odeljku bice navedeni oni elementi sistema koje je potrebno testirati. Ovde ce biti opisano samo sta je potrebno testirati, dok ce detalji o samim testovima biti navedeni kasnije u dokumentu test specifikacija.

##### **Testiranje podataka i integriteta base podataka**

Proveriti mogucnosti base ResiMe.

Proveriti mogucnost istovremenog pristupa istim podacima radi citanja.

Proveriti zakljucavanje podataka tokom pristupa bazi radi azuriranja.

##### **Testiranje sistema**

Proveriti slucaj koriscenja Izbor jezika.

Proveriti slucaj koriscenja pregled osnovnih informacija o laboratoriji.

Proveriti slucaj koriscenja pregled spiska clanova.

Proveriti slucaj koriscenja pregled podataka o odredjenom clanu laboratorije.

Proveriti slucaj koriscenja pregled publikacija po autoru

Proveriti slucaj koriscenja pregled publikacija po tipu

Proveriti slucaj koriscenja pregled publikacija po godini.

Proveriti slucaj koriscenja pregled spiska projekata

Proveriti slucaj koriscenja pregled podataka o odredjenom projektu

Proveriti slucaj koriscenja prijavljivanje.

Proveriti slucaj koriscenja azuriranje podataka o clanu.

Proveriti slucaj koriscenja dodavanje nove publikacije.

Proveriti slucaj koriscenja brisanje postojece publikacije.

Proveriti slucaj koriscenja azuriranje osnovnih podataka o laboratoriji.

Proveriti slucaj koriscenja kreiranje novog clana.

Proveriti slucaj koriscenja brisanje postojeceg clana.

Proveriti slucaj koriscenja arhiviranje postojeceg clana.

Proveriti slucaj koriscenja kreiranje projekta i postavljanje vodje.

Proveriti slucaj koriscenja azuriranje podataka o projektu.

Severska komponenta sistema treba da funkcioniše na racunaru pod operativnim sistemom Linux.

Severska komponenta sistema treba da funkcioniše na racunaru pod operativnim sistemom Windows.

Klijentska komponenta sistema treba da funkcioniše na racunaru Pentium I sa 128MB operativne memorije (Specifikacija zahteva, odeljak 7.6).

#### **Testiranje poslovnog ciklusa**

Proveriti posledice operacije dodavanja novog clana laboratorije.

Proveriti posledice operacije arhiviranja postojeceg clana laboratorije.

Proveriti posledice operacije definisanja novog projekta.

#### **Testiranje korisnickog interfejsa**

Proveriti jednostavnost navigacije kroz skup prozora aplikacije.

Proveriti da li izgled prozora aplikacije odgovara definisanim standardima za izradu korisnickog interfejsa.

Proveriti da li korisnicki interfejs omogucava jednostavno i intuitivno koriscenje bez potrebe za organizovanjem dodatne obuke.

Sistem treba da bude lak za koriscenje i prilagodjen korisnicima koji poseduju dobar nivo znanja rada na racunaru.

Proveriti da li online uputstvo obezbedjuje adekvatnu podrsku pri koriscenju naprednijih funkcionalnosti sistema.

#### **Testiranje performansi**

Proveriti vreme potrebno za prijavljivanje na sistem.

Proveriti vreme potrebno za dodavanje novog projekta.

Proveriti vreme potrebno za dodavanje publikacije.

Proveriti da li vreme potrebno za pristupanje bazi podataka u cilju izvršenje upita ne prelazi 5 sekundi.

#### **Testiranje opterećenja**

Proveriti odziv sistema kada mu istovremeno pristupa 200 posetilaca.

Proveriti odziv sistema kada 50 posetilaca istovremeno pristupa stranici pregled spiska projekata.

Proveriti odziv sistema kada 50 posetilaca istovremeno pristupa stranici pregled publikacija po autoru.

Proveriti da li sistem moze da podrzi do 1000 simultanih pristupa korisnika portalu.

#### **Testiranje kritичnih slučaja**

Proveriti vreme odziv sistema prilikom prvog koriscenja.

#### **Testiranje volumena**

Proveriti odziv sistema kada je 90% kapaciteta diska popunjeno.

#### **Testiranje sigurnosti i kontrole pristupa**

Proveriti mogućnost prijavljivanja na sistem sa racunara u lokalnoj mrezi.

Proveriti mogućnost prijavljivanja na sistem sa udaljenog racunara.

Proveriti prava pristupa korisnika koji pripadaju različitim grupama.

#### **Testiranje otkaza i oporavka**

Proveriti da li je ResiMe portal dostupan 24 casa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da predje 10%.

Proveriti da li srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.

#### **Testiranje konfiguracije**

Severska komponenta sistema treba da funkcioniše na racunaru pod operativnim sistemom Linux.

Severska komponenta sistema treba da funkcioniše na racunaru pod operativnim sistemom Windows.

Klijentska komponenta sistema treba da funkcioniše na racunaru Pentium I sa 128MB operativne memorije.

Proveriti da li klijentski deo ResiMe portala moze da se koristi pomocu Web citaca.

#### **Testiranje instalacije**

Proveriti instalaciju sistema na serveru.

### **5. Strategije testiranja**

Strategije testiranja predstavljaju preporuke na koji način treba vršiti testiranje softverske aplikacije. Prethodni odeljak o zahtevima testiranja opisuje šta će biti testirano, dok će u ovom odeljku biti opisano na kako će navedeni elementi biti testirani.

Osnova svake strategije testiranja jesu tehnike koje se koriste i kriterijumi na osnovu kojih se može utvrditi kada je testiranje završeno.

#### **5.1 Tipovi testiranja**

##### **5.1.1 Testiranje podataka i integriteta baze podataka**

Cilj testa:

Utvrđiti da metodi i procesi za pristup bazi podataka funkcionisu korektno i bez gresaka po podacima.

Tehnike:

- Probati svaki metod pristupa bazi podataka, za korektne i nekorektne podatke (ili za pribavljanje podataka)
- Ispitati bazu podataka da bi se utvrdilo da se podaci unose kao sto se ocekuje, da svi dogadjaji u bazi su obradjuju korektno i da se podaci pribavljaju korektno i bez gresaka.

Kriterijum zavrsetka:

Metodi i procesi za pristup bazama podataka rade kao sto je predvidjeno i bez gresaka u podacima.

Napomene:

- Testiranje zahteva koriscenje okruzenja za rad sa konkretnim DBMS-om da bi se direktno uneli i promenili podaci u bazi podataka.
- Procesi moraju da se rucno pokrenu.
- Koristiti bazu sa malim ili minimalnim brojem podataka, kako bi se povecala preglednost i lakse uocili neprihvatljivi slucajevi.

#### 5.1.2 Testiranje sistema

Cilj testiranja:

Potvrditi korektno kretanje kroz aplikaciju, unos podataka, njihovu obradu i pribavljanje.

Tehnike:

- Izvrsiti svaki slucaj koriscenja, ispratiti sve tokove definisane slucajevima koriscenja, koriscenjem tacnih i netacnih podataka.
- Utvrditi da li se ocekivani rezultati javljaju kada se radi sa tacnim podacima.
- Utvrditi da li se predvidjene poruke o gresci ili upozorenja javljaju kada se radi sa netacnim podacima.
- Potvrditi da su poslovna pravila pravilno primenjena.

Kriterijum zavrsetka:

- Svi planirani testovi su izvršeni.

Napomene:

- Potrebno je obezbediti pristup serveru na kome ce biti instaliran sistem.

#### 5.1.3 Testiranje poslovnog ciklusa

Cilj testiranja:

Potvrditi odgovarajucu primenu i procese koji se obavljaju kao podrška zahtevanom modelu poslovanja i utvrdjenom rasporedu.

Tehnike:

- Testiranje simulira poslovne cikluse koji obavljaju sledece:

- Testovi korisnici za testiranje funkcija aplikacija treba modifikovati/prosiriti da uvecaju broj izvršenja svake funkcije kako bi se simulirao pristup većeg broja različitih korisnika u odgovarajućem periodu vremena.
- Sve funkcije koje su vremenski zavisne treba proveriti koriscenjem tacnih i netacnih podataka.
- Proveriti da li se sve funkcije koje se pojavljuju periodicno da se izvrsavaju/pokrecu u odgovarajućem vremenskim trenucima.
- Utvrditi da li se ocekivani rezultati javljaju kada se radi sa tacnim podacima.
- Utvrditi da li se predvidjene poruke o gresci ili upozorenja javljaju kada se radi sa netacnim podacima.
- Potvrditi da su poslovna pravila pravilno primenjena.

Kriterijumi zavrsetka:

- Svi planirani testovi su izvršeni.
- Svi identifikovane greske su adekvatno obradjene.

Napomene:

- Odredjivanje datuma i perioda moze zahtevati dodatne aktivnosti.
- Poslovni model bi trebao da identifikuje odgovarajuće zahteve i procedure koje treba testirati.

#### 5.1.4 Testiranje korisnickog interfejsa

Cilj testiranja:

Potvrditi sledece:

- Navigacija kroz aplikaciju korektno odslikava poslovne funkcije i zahteve, ukljucujuci od kretanje izmedju prozora, polja i metode pristupa.
- Objekti prozora i njihove karakteristike, kao sto su meniji, velicina, pozicije, stanja i dobijanje fokusa u skladu sa standardima.

Tehnike:

- Kreirati/modifikovati testove za svaki prozor da bi se potvrdila odgovarajuća navigacija i stanja objekata za svaki prozor i objekat aplikacije.

Kriterijum zavrsetka:

Svaki prozor treba da se potvrdi da je konzistentan sa referentnom verzijom ili prihvacenim standardima.

Napomene:

#### 5.1.5 Testiranje performansi

Cilj testiranja:

Potvrditi vreme odziva sistema za napravljene transakcije ili poslovne funkcije za sledeca dva slucaja:

- procenjeno prosečno opterećenje
- procenjeno maksimalno opterećenje

Tehnike:

- Koristiti test procedure napravljene za testiranje poslovnog modela



- Modifikovati podatke tako da se uveća broj transakcija i modifikovati skripte da se poveća broj izvršenja transakcija
- Skripte treba da se izvršavaju na jednom racunaru (jedan korisnik, jedna transakcija) ili ponoviti sa više klijenata (virtuelnih ili stvarnih).

Kriterijum zavrsetka:

- Jedna transakcija / jedan korisnik: Uspesno izvršavanje test skripti bez ikakvih padova i za očekivano vreme.
- Više transakcija / više korisnika: Uspesno izvršavanje test skripti bez ikakvih padova i za očekivano vreme.

Napomene:

- Testiranje performansi sistema uključuje opterećenje sistema u pozadini. Postoji više metoda koji se mogu koristiti da bi se dobili zahtevani uslovi:
  - Pozivati transakcije direktno na serveru, obično u formi SQL poziva.
  - Kreirati virtuelnog korisnika koji će simulirati više klijenata. Može se iskoristiti softver koji će sa udaljenog racunara biti iskoriscen da poveća opterećenje. Time se, takodje, može postići povećanje opterećenja na mreži
  - Koristiti više ljudi koji će sa različitih racunara pokretati test skripte da bi se dostiglo odgovarajuće opterećenje.
- Testiranje performansi treba da se obavi na za to spremljenom racunaru u predviđeno vreme. Time se omogućuje potpuna kontrola i precizno merenje.
- Baza podataka koja se koristi za testiranje performansi treba da ima veličinu baze koja će se realno koristiti.

#### 5.1.6 Testiranje opterećenja

Cilj testiranja:

Potvrditi vreme odziva sistema za napravljene transakcije i poslovne slučajeve pod uslovima različitog opterećenja.

Tehnike:

- Koristiti testove napravljene za testiranje poslovnog ciklusa.
- Modifikovati podatke tako da se uveća broj transakcija i modifikovati skripte da se poveća broj izvršenja transakcija.

Kriterijum zavrsetka:

- Više transakcija / više korisnika: Uspesno izvršavanje test skripti bez ikakvih padova i za očekivano vreme

Napomene:

- Testiranje performansi treba da se obavi na za to spremljenom racunaru u predviđeno vreme. Time se omogućuje potpuna kontrola i precizno merenje.
- Baza podataka koja se koristi za testiranje performansi treba da ima veličinu baze koja će se realno koristiti.

#### 5.1.7 Testiranje kritičnih slučajeva

Cilj testiranja:

Proveriti da li sistem funkcioniše korektno i bez gresaka u sledecim kritичnim slucajevima:

- malo ili nimalo memorije na serveru
- maksimalan broj klijenata (stvarnih ili simuliranih) koji rade na sistemu
- vise korisnika koji zahtevaju istu transakciju nad istim podacima.
- najgori slucaj broja transakcija koje sistem moze da izdrzi.

Napomena: Cilj testiranja kritичnih slucajeva moze biti i odredjivanje uslova pod kojima sistem nije u mogucnosti da nastavi da radi korektno.

Tehnike:

- Koristiti testove koji su napravljeni za testiranje performansi.
- Da bi se obavilo testiranje pod ogranicenjima, testovi treba da se obalaju sa jednog racunara, a velicinu RAM i spoljne memorije treba smanjiti.
- Za ostale testove, treba koristiti vise klijenata, koji ce raditi na istom testu ili na komplementarnim testovima da bi se proizveo slucaj sa najvećom kolicinom podataka u transakciji.

Kriterijum zavrsetka:

Svi planirani testovi se izvrsavaju i predviđene granice sistema su dostignute bez pada sistema ili su uslovi pod kojima se pojavljuje otkaz sistema su izvan predviđenih granica.

Napomene:

- Testiranje mreze moze zahtevati koriscenje alata koji omogucavaju povećanje protoka podataka na mrezi.
- Velicina spoljne memorije treba da bude privremeno smanjena kako bi smanjila prostor koji moze da koristi baza podataka.
- Testiranje zahteva sinhronizaciju istovremenog pristupa klijenata istim slogovima.

#### 5.1.8 Testiranje volumena

Cilj testiranja:

Potvrditi da sistem funkcioniše uspesno u sledecim slucajevima:

- Maksimalan broj klijenata koji su povezani (simulirani) i koji izvrsavaju iste poslovne funkcije koje cine najgori slucaj za duzi period vremena.
- Maksimalna velicina baze podataka je dostignuta i vise transakcija koje izvrsavaju upite i pribavljaju podatke istovremeno.

Tehnike:

- Koristiti testove koji su napravljeni za testiranje performansi.
- Koristiti vise klijenata, koji ce raditi na istom testu ili na komplementarnim testovima da bi se proizveo slucaj sa najvećom kolicinom podataka u transakciji na duzi period vremena.
- Kreira se baza podataka maksimalne velicine (stvarna, uvecana ili napunjena reprezentativnim podacima) i vise klijenata koji pokrecu transakcije radi istovremenog pribavljanja podataka u toku duzeg perioda vremena.

Kriterijum zavrsetka:

- Svi planirani testovi se izvrsavaju i predviđene granice sistema su dostignute bez pada sistema.

Napomene:

- Odabrati period vremena koji ce biti prihvatljiv za testiranje uslova velikog volumena.

#### 5.1.9 Testiranje sigurnosti i kontrole pristupa

Cilj testiranja:

- Sigurnost funkcija/podataka: Potvrditi da korisnik moze da koristi samo one funkcije/ podatke koje su predvidjene za tip korisnika kome pripada.
- Sigurnost sistema: Potvrditi da samo oni korisnici kojima je dozvoljeno da pristupe sistemu mogu da pristupe sistemu.

Tehnike:

- Sigurnost funkcija/podataka: Identifikovati funkcije/podatke kojima ima pravo pristupa odgovarajuci tip korisnika
- Napraviti testove za svaki tip korisnika i potvrditi dozvolu da izvsava transakcije koje su specificne za taj tip korisnika.
- Izmeniti tip korisnika i proveriti ga na iste testove. Proveriti da li su funkcije /podaci korektno dozvoljeni ili zabranjeni za koriscenje.

Kriterijum zavrsetka:

- Za svaki tip korisnika i odgovarajuće funkcije/podatke dozvoljene za koriscenje, sve transakcije funkcionisu na ocekivani nacin, definisane u testovima sistema.

Napomene:

#### 5.1.10 Testiranje otkaza i oporavka

Cilj testiranja:

Potvrditi da proces opravka (rucni ili automatski) korektno vraća bazu podataka, aplikaciju i sistem u ocekivano stanje. Sledeci tipovi uslova treba da budu ukljuceni u testiranje:

- Iskljucenje napajanja na klijentu
- Iskljucenje napajanja na serveru
- Prekid komunikacije kroz mreznu
- Prekid komunikacije ili gubitak podataka usled greske rada hard diska
- Nekompletni ciklus
- Netacni pokazivaci (ključevi) u bazi podataka
- Netacni ili podatak u bazi podataka

Tehnike:

Test kreiran za testiranje funkcionalnosti i poslovnog ciklusa se koriste za formiranje niza transakcija. Kada se postigne pocetna tacka koja omogucava testiranje, sledece akcije treba da sprovedi (simulirati), individualno:

- Prekid napajanja na klijentu
- Prekid napajanja na serveru

- Prekid na mrezi: simulirati ili izazvati gubitak komunikacije kroz mrežu
- Prekid komunikacije, odnosno gubitak napajanja, sa hard diskom: simulirati ili fizički eliminisati komunikaciju sa jednim ili više kontrolera ili uređaja.

Kriterijumi završetka:

U svim opisanim slučajevima aplikacija, baza podataka i sistem treba, da se nakon završetka i procedura oporavka, vrate u očekivano stanje. To stanje uključuje da je naručavanje podataka ograničeno samo na narušena polja, pokazivače, ključeve i izveštaje u okviru procesa koji nisu završeni usled nastanka prekida.

Napomena:

- Testiranje oporavka zahteva operacije koje nekada nisu izvodive ili h nije poželjno obavljati (simulacija prekida komunikacije ili napajanja). Alternativa tome su softverski alati za dijagnostiku.
- Resursi sistema (racunara), baze podataka i mrežne grupe se koriste.
- Testiranje treba obavljati u casovima kada se resursi ne koriste ili na izolovanoj masini.

#### 5.1.11 Testiranje konfiguracije

Cilj testiranja:

Proveriti i potvrditi da klijentska aplikacija funkcioniše na konfiguracijama na kojima je to predviđeno.

Tehnike:

- Koristi testove integracije i funkcionalne testove.
- Pokrenuti/zatvoriti različite aplikacije, bilo pre pokretanja testa, bilo kao deo testiranja.
- Izvršiti izabrane transakcije da bi se simulirale korisničke aktivnosti u različitim aplikacijama.
- Ponoviti navedene procese, smanjujući RAM memoriju koju koristi klijent.

Kriterijum završetka:

Za svaku kombinaciju, transakcije iz testa treba da se završe uspešno, bez gresaka.

Napomene:

- Koje su aplikacije dostupne i mogu se koristiti na klijentu?
- Koje se aplikacije tipično koriste?
- Koje podatke aplikacija koristi
- Čeo sistem, mrežni serveri, baze podataka i slično treba biti opisano ka deo ovog testa.

#### 5.1.12 Testiranje instalacije

Cilj testiranja:

Proveriti i potvrditi da se softver korektno može instalirati na svakom od klijenata pod sledećim uslovima:

- Nova instalacija na novom računaru, nikad instalirana aplikacija.
- Azurirati računar prethodno instaliran sa istom verzijom aplikacije
- Azurirati računar prethodno instaliran sa prethodnom verzijom aplikacije

Tehnike:

- Ručno ili definisanjem automatskih skripti proveriti uslove na ciljnoj masini (nikad instalirana aplikacija, ista verzija, nova verzija).

- Pokrenuti i instalirati aplikaciju.
- Koriscenjem predefinisanih podskupa testova funkcionalnih testova, proveriti rad aplikacije.

Kriterijum zavrsetka:

Aplikacija treba da se izvrsavaju uspesno i bez gresaka.

Napomene:

- Koje transakcije treba izabrati da bi se napravio test koji ce sa pouzdanoscu potvrditi da je aplikacija uspesno instalirana i da ne fale pojedine softverske komponente?

## 5.2 Alati

Koristimo Sql Server Express 2022, Azure data studio, Visual studio code, Sql LocalDB i pretrazivac

## 6. Resursi

### 6.1 Radnici

Radnik	Preporuceni broj radnika	Posebne preporuke
Rukovodilac testiranja	/	Obezbedjuje nadgledanje upravljanjem procesa  Odgovornosti: <ul style="list-style-type: none"> <li>• Obezbediti tehnicku direkciju</li> <li>• Prikupljanje odgovarajucih resursa</li> <li>• Upravljanje izvestavanjem</li> </ul>
Projekant testova	/	Identifikuje, odredjuje prioritete i implementira slucajeve testiranja  Odgovornosti: <ul style="list-style-type: none"> <li>• Napraviti plan testiranja</li> <li>• Napraviti model testiranja</li> </ul> Proceniti efektivnost napora ulozenog u testiranje.
Tester sistema	/	Izvrstiti testiranje  Odgovornosti: <ul style="list-style-type: none"> <li>• Izvrstiti testiranje</li> <li>• Zabeleziti rezultate</li> <li>• Popraviti greske</li> </ul> Opisati greske
Administrator testiranja sistema	/	Pripremiti okruzenje za testiranje i utvrditi da ono odgovara pretpostavkama.  Odgovornosti:

		<ul style="list-style-type: none"> <li>Administrirati upravljanje testiranjem</li> </ul> <p>Upravlјati pristupom radnika resursima na kojima se obavlja testiranje.</p>
Administrator baze podataka	/	<p>Pripremiti podatke i bazu podataka za testiranje i utvrditi da njihovo stanje odgovara pretpostavkama.</p> <p>Odgovornosti:</p> <ul style="list-style-type: none"> <li>Administrirati podatke (bazu podataka) za testiranje</li> </ul>
Projektant	/	<p>Identifikovati i definisati operacije, atribute i veze izmedju razlicitih klasa testova</p> <p>Odgovornosti:</p> <ul style="list-style-type: none"> <li>Identifikovati i definisati klase testova</li> </ul> <p>Identifikovati i definisati pakete testova</p>
Implementacija	/	<p>Implementirati testove i rasporediti ih u odgovarajuće klase i pakete testova.</p> <p>Odgovornosti:</p> <ul style="list-style-type: none"> <li>Kreirati klase i pakete testova implementirane u test modelu.</li> </ul>

## 6.2 Sistem

Resurs	Ime/tip/serijski broj
Server	/
Klijentski test racunari	/
Lokalni racunari	/
Test repozitorijum	/
Racunari za razvoj testova	/
Simulator opterecenja	/

## 7. Kontrolne tacke testiranja

Zadatak	Trud	Pocetak	Kraj
Planiranje testova	/	/	/
Projektovanje testova			
Razvoj testova			
Izvršenje testova			
Evaluacija testova			

## 8. Izlazni produkti testiranja

### 8.1 Model testiranja

Model testiranja definise sve test slucajeve zajedno sa referencama na test procedure i test skripte koje su vezane za svaki test slucaj.

## 9. Projektni zadaci

### Plan testiranja

- Identifikovati zahteve za testiranjem
- Proceniti rizike
- Razviti strategije testiranja
- Identifikovati resurse testiranja
- Napraviti raspored
- Napraviti Plan testiranja

### Projektovanje testova

- Analizirati potrebni rad
- Razviti model testiranja
- Identifikovati i opisati test slucajeve
- Identifikovati i grupisati procedure testiranja
- Oceniti napraviti uvid u pokrivenost testiranja

### Implementacija testova

- Postaviti okruzenje za testiranje
- Napisati skripte za testiranje
- Razviti podrsku za testiranje
- Identifikovati specifne funkcionalnosti za projektovanje i implementaciju modela testiranja
- Utvrditi eksterne skupove podataka

### Testiranje

- Izvršiti test procedure
- Oceniti izvršenje testova
- Popraviti testove koji nisu prošli
- Proveriti rezultate
- Istraziti neočekivane rezultate
- Zabeležiti greske

#### **Ocena testiranja**

- Oceniti pokrivenost funkcionalnosti test slučajevima
- Proceniti pokrivenost koda test slučajevima
- Analizirati greske
- Utvrditi da li su kriterijumi zavrsetka i kriterijumi uspesnosti postignuti
- Napraviti Izveštaj o oceni testiranja