**CORE Team**

**LeProg**

**Plan testiranja**

**Verzija 1.0**

Istorija revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 10.02.2022 | 1.0 | Inicijalna verzija | Dženis |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Sadržaj**

[**1.** **Ciljevi** 4](#_Toc95393650)

[**2.** **Opseg** 4](#_Toc95393651)

[**3.** **Reference** 4](#_Toc95393652)

[**4.** **Zahtevi za testiranje** 4](#_Toc95393653)

[**5.** **Strategije testiranja (Test Strategy)** 7](#_Toc95393654)

[**5.1** **Tipovi testiranja (Testing Types)** 7](#_Toc95393655)

[5.1.1 Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing) 7](#_Toc95393656)

[5.1.2 Testiranje sistema (System Testing) 8](#_Toc95393657)

[5.1.3 Testiranje poslovnog ciklusa (Business Cycle Testing) 8](#_Toc95393658)

[5.1.4 Testiranje korisničkog interfejsa (User Interface Testing) 9](#_Toc95393659)

[5.1.5 Testiranje performansi (Performance Testing) 9](#_Toc95393660)

[5.1.6 Testiranje opterećenja (Load Testing) 10](#_Toc95393661)

[5.1.7 Testiranje kritičnih slučajeva(Stress Testing) 11](#_Toc95393662)

[5.1.8 Testiranje volumena (Volume Testing) 12](#_Toc95393663)

[5.1.9 Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing) 12](#_Toc95393664)

[5.1.10 Testiranje otkaza i oporavka (Failover / Recovery Testing) 13](#_Toc95393665)

[5.1.11 Testiranje konfiguracije (Configuration Testing) 14](#_Toc95393666)

[5.1.12 Testiranje instalacije (Installation Testing) 14](#_Toc95393667)

[**5.2**  **Alati** 15](#_Toc95393668)

[**6.** **Resursi** 16](#_Toc95393669)

[**6.1**  **Radnici** 16](#_Toc95393670)

[**6.2** **Sistem** 17](#_Toc95393671)

[**7.** **Kontrolne tačke testiranja** 18](#_Toc95393672)

[**8.** **Izlazni produkti testiranja** 18](#_Toc95393673)

[**8.1** **Model testiranja** 18](#_Toc95393674)

[**9.** **Projektni zadaci** 19](#_Toc95393675)

Plan testiranja

# **Ciljevi**

Ovaj dokument sadrži opis plana testiranja sistema LeProg. Namena ovog dokumenta je da opiše način provere funkcionalni i nefunkcionalne zahteve postavljene pri realizaciji predložene aplikacije. Ciljevi ovog dokumenta su:

* Izdvojiti komponente sistema koje treba testirati na osnovu postojećih informacija o projektu.
* Izdvojiti zahteve testiranja na osnovu datih predloga (na višem nivou).
* Predložiti i opisati strategije testiranja koje će biti korišćene.
* Odrediti resurse koji će biti potrebni za testiranje i proceniti vreme i trud koji treba uložiti pri testiranje.
* Navesti dokumente koji će opisati rezultate testiranja.

# **Opseg**

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

Plan testiranja ne opisuje testiranje pojedinačnih komponenti sistema, jer se pretpostavlja se da je svaki od programera izvršio kompletno testiranje koda pre njegovog dodavanja u sistem, korišćenjem metode crne kutije.

Plan testiranja će se odnositi na:

* testiranje interfejsa prema različitim tipovima korisnika,
* testiranje komunikacije sa bazom podataka,
* testiranje komunikacije komponenti aplikacije,
* testiranje funkcionalnosti izvršne verzije aplikacije i
* testiranje kritičnih performansi aplikacije.

# **Reference**

Reference na osnovu kojih će biti napisan Plan testiranja su:

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

# **4. Zahtevi za testiranje**

U ovom odeljku biće navedeni oni elementi sistema koje je potrebno testirati (slučajevi korišćenja, funkcionalni i nefunkcionalni zahtevi). Ovde će biti opisano samo šta je potrebno testirati, dok će detalji o samim testovima biti navedeni kasnije u dokumentu Test specifikacija.

**Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing)**

Proveriti mogućnost pristupa bazi LeProg.

Proveriti mogućnost istovremenog pristupa istim podacima radi čitanja.

Proveriti zaključavanje podataka tokom pristupa bazi radi ažuriranja.

Proveriti tačnost podataka koji se pribavljaju iz baze.

**Testiranje sistema – funkcionalno testiranje (System Testing – functional testing)**

Proveriti slučaj korišćenja Pregled broja korisnika koji su kupili kurs.

Proveriti slučaj korišćenja Ocenjivanje kursa.

Proveriti slučaj korišćenja Kupovina kurseva.

Proveriti slučaj korišćenja Pregled informacija vezanih za kurseve.

Proveriti slučaj korišćenja Čitanje članaka.

Proveriti slučaj korišćenja Pregled ocena nekog kursa.

Proveriti slučaj korišćenja Registrovanje.

Proveriti slučaj korišćenja Odjavljivanje.

Proveriti slučaj korišćenja Promena lozinke.

Proveriti slučaj korišćenja Prijavljivanje.

Proveriti slučaj korišćenja Postavljanje kursa.

Proveriti slučaj korišćenja Uklanjanje kursa.

Proveriti slučaj korišćenja Pregledanje spiska korisnika.

Proveriti slučaj korišćenja Prihvatanje korisnika.

Proveriti slučaj korišćenja Pregled podataka o određenom korisniku.

Proveriti slučaj korišćenja Brisanje korisnika.

Proveriti slučaj korišćenja Dodavanje članka.

Proveriti slučaj korišćenja Uklanjanje članka.

Proveriti slučaj korišćenja Ažuriranje članka.

Proveriti slučaj korišćenja Uklanjanje iz liste želja.

Proveriti slučaj korišćenja Pregled liste želja.

Proveriti slučaj korišćenja Slanje zahteva.

Proveriti slučaj korišćenja Dodavanje u listu želja.

Proveriti slučaj korišćenja Pretraga kursa na osnovu cene.

Proveriti slučaj korišćenja Pretraga kursa na osnovu ocene.

Proveriti slučaj korišćenja Pretraga kursa na osnovu levela.

Proveriti slučaj korišćenja Pretraga kursa na osnovu oblasti.

Proveriti slučaj korišćenja Pretraga kursa na osnovu imena.

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

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

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

**Testiranje poslovnog ciklusa (Business Cycle Testing)**

Proveriti posledice operacije dodavanja novog člana.

Proveriti posledice operacije arhiviranja postojećeg člana.

Proveriti posledice operacije definisanja novog projekta.

**Testiranje korisničkog interfejsa (User Interface Testing)**

Proveriti jednostavnost navigacije kroz skup prozora aplikacije.

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

Proveriti da li korisnički interfejs omogućava jednostavno i intuitivno korišćenje bez potrebe za organizovanjem dodatne obuke (Specifikacija zahteva, odeljak 7.2).

Sistem treba da bude lak za korišćenje i prilagođen korisnicima koji poseduju dobar nivo znanja rada na računaru (Vizija sistema, odeljak 5.2).

Proveriti da li *online* uputstvo obezbeđuje adekvatnu podršku pri korišćenju naprednijih funkcionalnosti sistema (Vizija sistema, odeljak 12.2).

**Testiranje performansi (Performance Testing)**

Proveriti vreme potrebno za prijavljivanje na sistem.

Proveriti vreme potrebno za dodavanje novog kursa.

Proveriti vreme potrebno za dodavanje članka.

Proveriti da li vreme potrebno za pristupanje bazi podataka u cilju izvršenja upita ne prelazi 5 sekundi (Specifikacija zahteva, odeljak 7.4).

**Testiranje opterećenja (Load Testing)**

Proveriti odziv sistema kada mu istovremeno pristupa 200 posetilaca.

Proveriti odziv sistema kada 100 posetilaca istovremeno pristupa stranici Pregled informacija o kursu.

Proveriti odziv sistema kada 50 posetilaca istovremeno pristupa stranici Pregled spiska kurseva.

Proveriti odziv sistema kada 50 posetilaca istovremeno pristupa stranici Pregled članaka.

Proveriti da li sistem može da podrži do 1000 simultanih pristupa korisnika portalu (Detaljni arhitekturni projekat, odeljak 11).

**Testiranje kritičnih slučajeva (Stress Testing)**

Proveriti vreme odziv sistema prilikom prvog korišćenja.

Proveriti vreme odziva sistema kada 100 posetilaca istovremeno pristupa stranici Pregled osnovnih podataka o kursu.

**Testiranje volumena (Volume Testing)**

Proveriti odziv sistema kada je 90% kapaciteta diska popunjeno.

**Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing)**

Proveriti mogućnost prijavljivanja na sistem sa računara u lokalnoj mreži.

Proveriti mogućnost prijavljivanja na sistem sa udaljenog računara.

Proveriti prava pristupa korisnika koji pripadaju različitim grupama (Anonimni korisnik,Registrovan korisnik, Predavač,Administrator).

**Testiranje otkaza i oporavka (Failover / Recovery Testing)**

Proveriti da li je Leprog portal dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%. (Vizija sistema, odeljak 9; Detaljni arhitekturni projekat, odeljak 12)

Proveriti da li srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati (Detaljni arhitekturni projekat, odeljak 12).

**Testiranje konfiguracije (Configuration Testing)**

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

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

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

Proveriti da li klijentski deo PeNcIL portala može da se koristi pomoću Web čitača: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, kao i Firefox (Mozilla) (Specifikacija sistema, odeljak 7.6).

**Testiranje instalacije (Installation Testing)**

Proveriti instalaciju sistema na serveru.

# **5. Strategije testiranja (Test Strategy)**

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 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 (Testing Types)**

### 5.1.1 Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing)

|  |  |
| --- | --- |
| Cilj Testa: | Utvrditi da metodi i procesi za pristup bazi podataka funkcionišu korektno i bez grešaka po podacima. |
| Tehnike: | 1. Probati svaki metod pristupa bazi podataka, za korektne i nekorektne podatke (ili za pribavljanje podataka) 2. Ispitati bazu podataka da bi se utvrdilo da se podaci unose kao što se očekuje, da svi događaji u bazi su obrađuju korektno i da se podaci pribavljaju korektno i bez grešaka. |
| Kriterijum završetka: | Metodi i procesi za pristup bazama podataka rade kao što je predviđeno i bez grešaka u podacima. |
| Napomene: | 1. Testiranje zahteva korišćenje okruženja za rad sa konkretnim DBMS-om da bi se direktno uneli i promenili podaci u bazi podataka. 2. Procesi moraju da se ručno pokrenu. 3. Koristiti bazu sa malim ili minimalnim brojem podataka, kako bi se povećala preglednost i lakše uočili neprihvatljivi slučajevi. |

### 5.1.2 Testiranje sistema (System Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi korektno kretanje kroz aplikaciju, unos podataka, njihovu obradu i pribavljanje. |
| Tehnike: | 1. Izvršiti svaki slučaj korišćenja, ispratiti sve tokove definisane slučajevima korišćenja, korišćenjem tačnih i netačnih podataka. 2. Utvrditi da li se očekivani rezultati javljaju kada se radi sa tačnim podacima. 3. Utvrditi da li se predviđene poruke o grešci ili upozorenja javljaju kada se radi sa netačnim podacima. 4. Potvrditi da su poslovna pravila pravilno primenjena. |
| Kriterijum završetka: | 1. Svi planirani testovi su izvršeni. 2. Svi identifikovane greške su adekvatno obrađene. |
| Napomene: | 1. Potrebno je obezbediti pristup serveru na kome će biti instaliran sistem |

### 5.1.3 Testiranje poslovnog ciklusa (Business Cycle Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi odgovarajuću primenu i procese koji se obavljaju kao podrška zahtevanom modelu poslovanja i utvrđenom rasporedu |
| Tehnike: | 1. Testiranje simulira poslovne cikluse koji obavljaju sledeće: 2. Testovi korišćeni za testiranje funkcija aplikacija treba modifikovati/proširiti da uvećaju broj izvršenja svake funkcije kako bi se simulirao pristup većeg broja različitih korisnika u odgovarajućem periodu vremena. 3. Sve funkcije koje su vremenski zavisne treba proveriti korišćenjem tačnih i netačnih podataka. 4. Proveriti da li se sve funkcije koje se pojavljuju periodično da se izvršavaju/pokreću u odgovarajućem vremenskim trenucima. 5. Utvrditi da li se očekivani rezultati javljaju kada se radi sa tačnim podacima. 6. Utvrditi da li se predviđene poruke o grešci ili upozorenja javljaju kada se radi sa netačnim podacima. 7. Potvrditi da su poslovna pravila pravilno primenjena. |
| Kriterijum završetka: | 1. Svi planirani testovi su izvršeni. 2. Svi identifikovane greške su adekvatno obrađene. |
| Napomene: | 1. Određivanje datuma i perioda može zahtevati dodatne aktivnosti. 2. Poslovni model bi trebao da identifikuje odgovarajuće zahteve i procedure koje treba testirati. |

### 5.1.4 Testiranje korisničkog interfejsa (User Interface Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi sledeće:   * Navigacija kroz aplikaciju korektno odslikava poslovne funkcije i zahteve, uključujući od kretanje između prozora, polja i metode pristupa (prelazak tabom, kretanje miša, prečice sa tastature) * Objekti prozora i njihove karakteristike, kao što su meniji, veličina, 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 završetka: | Svaki prozor treba da se potvrdi da je konzistentan sa referentnom verzijom ili prihvaćenim standardima. |
| Napomene: |  |

### 5.1.5 Testiranje performansi (Performance Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi vreme odziva sistema za napravljene transakcije ili poslovne funkcije za sledeća dva slučaja:  - procenjeno prosečno opterećenje  - procenjeno maksimalno opterećenje |
| Tehnike: | 1. Koristiti test procedure napravljene za testiranje poslovnog modela (testiranje sistema) 2. Modifikovati podatke tako da se uveća broj transakcija i modifikovati skripte da se poveća broj izvršenja transakcija. 3. Skripte treba da se izvršavaju na jednom računaru (jedan korisnik, jedna transakcija) ili ponoviti sa više klijenata (viruelinih ili stvarnih). |
| Kriterijum završetka: | 1. Jedna transakcija / jedan korisnik: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme. 2. Više transakcija / više korisnika: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme. |
| Napomena: | 1. Testiranje performansi sistema uključuje opterećenje sistema u pozadini. Postoji više metoda koji se mogu koristiti da bi se dobili zahtevani uslovi: 2. Pozivati transakcije direktno na serveru, obično u formi SQL poziva. 3. Kreirati virtuelnog korisnika koji će simulirati više klijenata. Može se iskoristiti softver koji će sa udaljenog računara biti iskorišćen da poveća opterećenje. Time se, takođe, može postići povećanje opterećenja na mreži. 4. Koristiti više ljudi koji će sa različitih računara pokretati test skripte da bi se dostiglo odgovarajuće opterećenje. 5. Testiranje performansi treba da se obavi na za to spremljenom računaru u predviđeno vreme. Time se omogućuje potpuna kontrola i precizno merenje. 6. Baza podataka koja se koristi za testiranje performansi treba da ima veličinu baze koja će se realno koristiti. |

### 5.1.6 Testiranje opterećenja (Load Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi vreme odziva sistema za napravljene transakcije i poslovne slučajeve pod uslovima različitog opterećenja. |
| Tehnike: | 1. Koristiti testove napravljene za testiranje poslovnog ciklusa. 2. Modifikovati podatke tako da se uveća broj transakcija i modifikovati skripte da se poveća broj izvršenja transakcija. |
| Kriterijum završetka: | 1. Više transakcija / više korisnika: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme. |
| Napomene: | 1. Testiranje performansi treba da se obavi na za to spremljenom računaru u predviđeno vreme. Time se omogućuje potpuna kontrola i precizno merenje. 2. Baza podataka koja se koristi za testiranje performansi treba da ima veličinu baze koja će se realno koristiti. |

### 5.1.7 Testiranje kritičnih slučajeva(Stress Testing)

|  |  |
| --- | --- |
| Cilj testa: | Proveriti da li sistem funkcioniše korektno i bez grešaka u sledećim kritičnim slučajevima:   * malo ili nimalo memorije na serveru * maksimalan broj klijenata (stvarnih ili simuliranih) koji rade na sistemu * više korisnika koji zahtevaju istu transakciju nad istim podacima. * najgori slučaj broja transakcija koje sistem može da izdrži.   Napmena: Cilj testiranja kritičnih slučajeva može biti i određivanje uslova pod kojima sistem nije u mogućnosti da nastavi da radi korektno. |
| Tehnike: | * Koristiti testove koji su napravljeni za testiranje performansi. * Da bi se obavilo testiranje pod ograničenjima, testovi treba da se obaljaju sa jednog računara, a veličinu RAM i spoljne memorije treba smanjiti. * Za ostale testove, treba koristiti više klijenata, koji će raditi na istom testu ili na komplementarnim testovima da bi se proizveo slučaj sa najvećim količinom podataka u transakciji. |
| Kriterijum završetka: | Svi planirani testovi se izvršavaju i predviđene granice sistema su dostignute bez pada sistema ili su uslovi pod kojima se pojavljuje otkaz sistema su izvan predviđenih granica. |
| Napomene: | * Testiranje mreže može zahtevati korišćenje alata koji omogućavaju povećanje protoka podataka na mreži. * Veličina spoljne memorije treba da bude privremeno smanjena kako bi smanjila prostor koji može da koristi baza podataka. * Testiranje zahteva sinhronizaciju istovremenog pristupa klijenata istim slogovima. |

### 5.1.8 Testiranje volumena (Volume Testing)

|  |  |
| --- | --- |
| Cilj testa: | Potvrditi da sistem funkcioniše uspešno u sledećim slučajevima:   * maksimalan () broj klijenata koji su povezani (simulirani) i koji izvršavaju iste poslovne funkcije koje čine najgori slučaj za duži period vremena. * Maksimalna veličina baze podataka je dostignuta i više transakcija koje izvršavaju upite i pribavljaju podatke istovremeno. |
| Tehnike: | * Koristiti testove koji su napravljeni za testiranje performansi. * Koristiti više klijenata, koji će raditi na istom testu ili na komplementarnim testovima da bi se proizveo slučaj sa najvećim količinom podataka u transakciji na duži period vremena. * Kreira se baza podataka maksimalne veličine (stvarna, uvećana ili napunjena reprezentativnim podacima) i više klijenata koji pokreću transakcije radi istovremenog pribavljanja podataka u toku dužeg perioda vremena. |
| Kriterijum završetka: | Svi planirani testovi se izvršavaju i predviđene granice sistema su dostignute bez pada sistema. |
| Napomene: | * Odabrati period vremena koji će biti prihvatljiv za testiranje uslova velikog volumena. |

### 5.1.9 Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Sigurnost funkcija/podataka: Potvrditi da korisnik može da koristi samo one funkcije/ podatke koje su predviđene za tip korisnika kome pripada.  Sigurnost sistema: Potvrditi da samo oni korisnici kojima je dozvoljeno da pristupe sistemu mogu da pristupe sistemu. |
| Tehnike: | 1. Sigurnost funkcija/podataka: Identifikovati funkcije/podatke kojima ima pravo pristupa odgovarajući tip korisnika. 2. Napraviti testove za svaki tip korisnika i potvrditi dozvolu da izvršava transakcije koje su specifične za taj tip korisnika. 3. Izmeniti tip korisnika i proveriti ga na iste testove. Proveriti da li su funkcije /podaci korektno dozvoljeni ili zabranjeni za korišćenje. |
| Kriterijum završetka: | Za svaki tip korisnika i odgovarajuće funkcije/podatke dozvoljene za korišćenje, sve transakcije funkcionišu na očekivani način, definisane u testovima sistema. |
| Napomene: |  |

### 5.1.10 Testiranje otkaza i oporavka (Failover / Recovery Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi da proces opravka (ručni ili automatski) korektno vraća bazu podataka, aplikaciju i sistem u očekivano stanje. Sledeći tipovi uslova treba da budu uključeni u testiranje:   1. Isključenje napajanja na klijentu 2. Isključenje napajanja na serveru 3. Prekid komunikacije kroz mrežnu 4. Prekid komunikacije ili gubitak podataka usled greške rada hard diska 5. Nekompletni ciklus (prekid u procesu filtriranja podataka, prekid u procesu sinhronizacije podataka) 6. Netačni pokazivači (ključevi) u bazi podataka 7. Netačni ili podatak u bazi podataka |
| Tehnike: | Test kreiran za testiranje funkcionalnosti i poslovnog ciklusa se koriste za formiranje niza transakcija. Kada se postigne početna tačka koja omogućava testiranje, sledeće akcije treba da sprovesti (simulirati), individualno:   1. Prekid napajanja na klijentu 2. Prekid napajanja na serveru 3. Prekid na mreži: simulirati ili izazvati gubitak komunikacije kroz mrežu (fizički isključiti komunikacioni kabl ili isključiti mrežni server ili ruter) 4. Prekid komunikacije, odnosno gubitak napajanja, sa hard diskom: simulirati ili fizički eliminisati komunikaciju sa jednim ili više kontrolera ili uređaja.   Kada se uslovi iznad ostvare, dodatne transakcije treba izvršiti i pre dostizanja naredne tačke testa, procedure za oporavak treba pokrenuti.  Testiranje nekompletnih ciklusa koristi istu tehniku opisanu iznad, s tim što procesi na bazi podataka treba da budu prekinuti ili zaustavljeni.  Testiranje na sledeće uslove zahteva da se sistem nalazi u poznatom stanju. Nekoliko polja u bazi podataka, pokazivača i ključeva treba narušiti ručno i direktno u bazi podataka (korišćenjem alata baze podataka). Dodatne transakcije treba izvršiti korišćenjem testova iz testiranja funkcionalnosti i testiranja poslovnog ciklusa. |
| Kriterijum 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 narušavanje podataka ograničeno samo na narušena polja, pokazivače, ključeve i izveštaje u okviru procesa koji nisu završeni usled nastanka prekida. |
| Napomena: | 1. 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. 2. Resursi sistema (računara), baze podataka i mrežne grupe se koriste. 3. Testiranje treba obavljati u časovima kada se resursi ne koriste ili na izolovanoj mašini. |

### 5.1.11 Testiranje konfiguracije (Configuration Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Proveriti i potvrditi da klijentska aplikacija funkcioniše na konfiguracijama na kojima je to predviđeno. |
| Tehnike: | 1. Koristi testove integracije i funkcionalne testove. 2. Pokrenuti/zatvoriti različite aplikacije, bilo pre pokretanja testa, bilo kao deo testiranja. 3. Izvršiti izabrane transakcije da bi se simulirale korisničke aktivnosti u različitim aplikacijama. 4. Ponoviti navedene procese, smanjujući RAM memoriju koju koristi klijent. |
| Kriterijum završetka: | Za svaku kombinaciju, transakcije iz testa treba da se završe uspešno, bez grešaka. |
| Napomene: | 1. Koje su aplikacije dostupne i mogu se koristiti na klijentu? 2. Koje se aplikacije tipično koriste? 3. Koje podatke aplikacija koristi (koji je obim tih podataka) 4. Ceo sistem, mrežni serveri, baze podataka i slično treba biti opisano ka deo ovog testa. |

### 5.1.12 Testiranje instalacije (Installation Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Proveriti i potvrditi da se softver korektno može instalirati na svakom od klijenata pod sledećim uslovima:   1. Nova instalacija na novom računaru, nikad instalirana aplikacija. 2. Ažurirati računar prethodno instaliran sa istom verzijom aplikacije 3. Ažurirati računar prethodno instaliran sa prethodnom verzijom aplikacije |
| Tehnike: | 1. Ručno ili definisanjem automatskih skripti proveriti uslove na ciljnoj mašini (nikad instalirana aplikacija, ista verzija, nova verzija). 2. Pokrenuti i instalirati aplikaciju. 3. Korišćenjem predefinisanog podskupa testova funkcionalnih testova, proveriti rad aplikacije. |
| Kriterijum završetka: | Aplikacija treba da se izvršavaju uspešno i bez grešaka. |
| Napomene: | 1. Koje transakcije treba izabrati da bi se napravio test koji će sa pouzdanošću potvrditi da je aplikacija uspešno instalirana i da ne fale pojedine softverske komponente? |

## **5.2 Alati**

U sledećoj tabeli su pobrojani alati koji će biti korišćeni za testiranje sistema:

|  |  |  |  |
| --- | --- | --- | --- |
| **Faza** | **Alat** | **Proizvođač/ matični proiz.** | **Verzija** |
| Test Management | Rational Unified Process | Rational | TBD |
| Test Design | Rational Rose | Rational | TBD |
| Defect Tracking |  |  | TBD |
| Functional Testing |  |  | TBD |
| Performance Testing |  |  | TBD |
| Test Coverage Monitor or Profiler |  |  | TBD |
| Other Test Tools |  |  | TBD |
| Project Management | Project  Word  Excel | Microsoft  Microsoft  Microsoft | TBD |
| DBMS tools | SQLite | SQLite | TBD |

# **6. Resursi**

U ovom odelju su opisani resursi koji se mogu koristiti za testiranje sistema, glavne odgovornosti i njihovo znanje ili veštine.

## **6.1 Radnici**

Ova tabela pokazuje predlog rasporeda radnika na aktivnostima testiranja.

|  |  |  |
| --- | --- | --- |
| **Ljudski resursi** | | |
| **Radnik** | **Preporučeni minimalni broj radnika**  (broj radnika koji će raditi puno radno vreme) | **Posebne preporuke / komentari** |
| Rukovodilac testiranja | Dženis, Aldina | Obezbeđuje nadgledanje upravljanjem procesa  Odgovornosti:   1. Obezbediti tehničku direkciju 2. Prikupljanje odgovarajućih resursa 3. Upravljanje izveštavanjem |
| Projektant testova | Dženis, Aldina | Identifikuje, određuje prioritete i implementira slučajeve testiranja  Odgovornosti:   1. Napraviti plan testiranja 2. Napraviti model testiranja 3. Proceniti efektivnost napora uloženog u testiranje. |
| Tester sistema | Aldina | Izvršiti testiranje  Odgovornosti:   1. Izvršiti testiranje 2. Zabeležiti rezultate 3. Popraviti greške 4. Opisati greške |
| Administrator testiranja sistema | Dženis | Pripremiti okruženje za testiranje i utvrditi da ono odgovara pretpostavkama.  Odgovornosti:   1. Administrirati upravljanje testiranjem 2. Upravljati pristupom radnika resursima na kojima se obavlja testiranje. |
| Administrator baze podataka | Dženis | Pripremiti podatke i bazu podataka za testiranje i utvrditi da njihovo stanje odgovara pretpostavkama.  Odgovornosti:   1. Administrirati podatke (bazu podataka) za testiranje |
| Projektant | Aldina | Identifikovati i definisati operacije, atribute i veze između različitih klasa testova  Odgovornosti:   1. Identifikovati i definisati klase testova 2. Identifikovati i definisati pakete testova |
| Implementacija | Dženis  Aldina | Implementirati testove i rasporediti ih u odgovarajuće klase i pakete testova.  Odgovornosti:   1. Kreirati klase i pakete testova implementirane u test modelu. |

## **6.2 Sistem**

U sledećoj tabeli su navedeni sistemski resursi koji će se koristiti za testiranje Leprog sistema.

|  |  |
| --- | --- |
| **Sistemski resursi** | |
| **Resurs** | **Ime / Tip / Serijski broj** |
| Server | Serijski broj: X179773562b |
| Klijentski Test Računari |  |
| —3 udaljena računara  (sa internet pristupom) | Serijski broj: A8339223  Serijski broj: B9334022  Serijski broj: B9332544 |
| —5 lokalnih računara  (povezanih u lokanu mrežu) | Serijski broj: R3322411 (IT Lab)  Serijski broj: A8832234 (IT Lab)  Serijski broj: W4592233 (IT Lab)  Serijski broj: X3333411 (IT Lab)  Serijski broj: X9834000 (Student Union) |
| Test repozitorijum |  |
| —Server | Serial No: X179773562b |
| Računari za razvoj testova - 6 | Serijski broj: R3322411  Serijski broj: A8832234  Serijski broj: W4592233  Serijski broj: X3333411 |
| Simulator opterećenja | Serijski broj: ABC-123 |

# **7. Kontrolne tačke testiranja**

Aktivnosti testiranja zavise od iteracija u razvoju softvera. Ciklus razvoja softvera Leprog biće odrađen u jednoj iteraciji koja sadrži kompletan ciklus testiranja, planiranje, projektovanje, razvoj, izvršenje i evaluaciju.

Sledeća table prikazuje kontrolne tačke testiranja.

|  |  |  |  |
| --- | --- | --- | --- |
| **Zadatak** | **Trud (pd)** | **Početak** | **Kraj** |
| Iteracija C1: R1.0 Release  Planiranje testova  Projektovanje testova  Razvoj testova  Izvršenje testova  Evaluacija testova | 5 | Februar 20 | Februar 27 |

# **8. Izlazni produkti testiranja**

Rezultati testiranja koji su definisani ovim planom testiranja navedeni su u sledećoj tabeli.

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokument** | **Vlasnik** | **Ocena / Raspodela** | **Due Date** |
| Plan Testiranja | Dženis | Rukovodioci testiranja | Maj 07 |
| Test okruženje | Dženis | - | Maj 07 |
| Model testiranja | Dženis i Aldina | Interna ocena | Maj 07 |
| Skup test podataka | Aldina | Interna ocena | Maj 08 |
| Test procedure | Dženis | Interna ocena | Maj 08 |
| Test skripte | Aldina | - | Maj 08 |
| Test podrška | Dženis | - | Maj 08 |
| Izveštaj o uočenim greškama | Dženis i Aldina | Rukovodioci testiranja | Maj 11 |
| Rezultati testa | Dženis | Rukovodilac testiranja | Maj 11 |
| Izveštaj o evaluaciji testa | Aldina | Rukovodilac testiranja | Maj 11 |

## **8.1 Model testiranja**

Model testiranja definiše sve test slučajeve zajedno sa referencama na test procedure i test skripte koje su vezane za svaki test slučaj.

# **9. Projektni zadaci**

U sledećoj tabeli su navedeni projektni zadaci testiranja sistema:

|  |
| --- |
| **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 slučajeve |
| Identifikovati i grupisati procedure testiranja |
| Oceniti napraviti uvid u pokrivenost testiranja |
| **Implementacija testova** |
| Postaviti okruženje za testiranje |
| Napisati skripte za testiranje |
| Razviti podršku za testiranje |
| Identifikovati sprecifične 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 |
| Istražiti neočekivane rezultate |
| Zabeležiti greške |
| **Ocena testiranja** |
| Oceniti pokrivenost funkcionalnosti test slučajevima |
| Proceniti pokrivenost koda test slučajevima |
| Analizirati greške |
| Utvrditi da li su kriterijumi završetka i kriterijumi uspešnosti postignuti |
| Napraviti Izveštaj o oceni testiranja |