Adiutor

Web portal za edukaciju

Verzija 1.0

Istorija revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 8.05.2016 | 1.0 | Inicijalna verzija | Mladenović Miloš,  Radosavljević Marko,  Nikolić Predrag,  Ranković Nikola |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Ciljevi 4

2. Opseg 4

3. Reference 4

4. Zahtevi za testiranje 4

5. Strategije testiranja (Test Strategy) 8

5.1 Tipovi testiranja (Testing Types) 8

5.1.1 Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing) 8

5.1.2 Testiranje sistema (System Testing) 8

5.1.3 Testiranje poslovnog ciklusa (Business Cycle Testing) 9

5.1.4 Testiranje korisničkog interfejsa (User Interface Testing) 9

5.1.5 Testiranje performansi (Performance Testing) 10

5.1.6 Testiranje opterećenja (Load Testing) 11

5.1.7 Testiranje kritičnih slučajeva(Stress Testing) 11

5.1.8 Testiranje volumena (Volume Testing) 12

5.1.9 Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing) 13

5.1.10 Testiranje otkaza i oporavka (Failover / Recovery Testing) 13

5.1.11 Testiranje konfiguracije (Configuration Testing) 14

5.1.12 Testiranje instalacije (Installation Testing) 15

5.2 Alati 15

6. Resursi 16

6.1 Radnici 16

6.2 Sistem 17

7. Kontrolne tačke testiranja 18

8. Izlazni produkti testiranja—treba mi ono odozgo da bi ovo napravio 18

8.1 Model testiranja 19

9. Projektni zadaci 19

Adiutor

# Ciljevi

Ovaj dokument sadrži opis plana testiranja Adiutor portala. Namena ovog dokumenta je da opiše način provere funkcionalnih i nefunkcionalnih zahteva postavljenih 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 Adiutor portala.

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 portala,
* testiranje funkcionalnosti izvršne verzije portala i
* testiranje kritičnih performansi portala.

# Reference

Reference na osnovu kojih će biti napisan Plan testiranja su:

1. Adiutor – Vizija sistema - V1.0 - 2016 – A4 Team
2. Adiutor – Specifikacija zahteva – V1.0 -2016 – A4 Team
3. Adiutor – Arhitekturni projekat – V1.1 – 2016 – A4 Team

# 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 Adiutor portala.

Proveriti mogućnost istovremenog pristupa istim podacima radi čitanja.

Proveriti zaključavanje podataka tokom pristupa bazi radi ažuriranja.

Proveriti očuvanje referencijalnog integriteta baze.

Proveriti tačnost podataka koji se pribavljaju iz baze.

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

**Provera slučajeva korišćenja**

1. Provera slučaja korišćenja Prijavljivanje
2. Provera slučaja korišćenja Registrovanje
3. Provera slučaja korišćenja Prijavljivanje na kurs
4. Provera slučaja korišćenja Informisanje studenata
5. Provera slučaja korišćenja Prihvatanje zahteva za pristup kursu
6. Provera slučaja korišćenja Arhiviranje kursa
7. Provera slučaja korišćenja Dodeljivanje profesora
8. Provera slučaja korišćenja Dodavanje kursa
9. Provera slučaja korišćenja Pregled liste naloga
10. Provera slučaja korišćenja Promena tipa naloga
11. Provera slučaja korišćenja Prihvatanje zahteva za registraciju
12. Provera slučaja korišćenja Unos poena
13. Provera slučaja korišćenja Ocenjivanje aktivnosti
14. Provera slučaja korišćenja Svođenje ocene
15. Provera slučaja korišćenja Postavljanje anketa
16. Provera slučaja korišćenja Pregled rezultata ankete
17. Provera slučaja korišćenja Postavljanje testova
18. Provera slučaja korišćenja Dodavanje oblasti
19. Provera slučaja korišćenja Arhiviranje oblasti
20. Provera slučaja korišćenja Dodavanje lekcije
21. Provera slučaja korišćenja Ažuriranje lekcije
22. Provera slučaja korišćenja Arhiviranje lekcije
23. Provera slučaja korišćenja Pregled liste kurseva
24. Provera slučaja korišćenja Pristup kursu
25. Provera slučaja korišćenja Pristup oblasti u okviru kursa
26. Provera slučaja korišćenja Pristup lekciji
27. Provera slučaja korišćenja Pregled liste slobodnih kurseva
28. Provera slučaja korišćenja Pregled obavestenja
29. Provera slučaja korišćenja Polaganje testova
30. Provera slučaja korišćenja Pregled liste svih foruma
31. Provera slučaja korišćenja Pregled liste tema
32. Provera slučaja korišćenja Čitanje postova
33. Provera slučaja korišćenja Postavljanje nove teme
34. Provera slučaja korišćenja Postavljanje postova
35. Provera slučaja korišćenja Pregled povezanih lekcija
36. Provera slučaja korišćenja Rangiranje
37. Provera slučaja korišćenja Unos kljucnih reci
38. Provera slučaja korišćenja Popunjavanje ankete
39. Provera slučaja korišćenja Ocenjivanje lekcije

**Provera dodatnih funkcionalnih zahteva sistema**

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 IV sa 512MB operativne memorije, kao i na telefonima sledeće konfiguracije ili bolje:

* Css resolution 240x320
* Pixel resolution 240x320
* Operating system Android 2.3.5, iOS, Windows Phone

(Specifikacija zahteva, odeljak 7.6).

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

Proveriti posledice operacija koje se sprovode u toku trajanja kursa(jednog semestra). To podrazumeva sledeće:

* Proveriti posledice operacije dodavanja kursa na portal.
* Proveriti posledice operacije registrovanja novog korisnika i prijave studenata za pohađanje novog kursa
* Proveriti posledice operacije dodeljivanja novog profesora na predmetu
* Proveriti posledice operacije dodavanja nove oblasti ili lekcije.
* Proveriti posledice operacije postavljanja testova u toku trajanja kursa, kao i njihove izrade.
* Proveriti posledice operacije postavljanja ankete na kraju kursa, kao i njene izrade.
* Proveriti posledice operacije svođenja ocena na kraju kursa.

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

Proveriti jednostavnost navigacije kroz skup prozora aplikacije.

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 osnovan nivo znanja rada na računaru (Vizija sistema, odeljak 5.1).

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

**Testiranje performansi (Performance Testing)**

Proveriti vreme potrebno za prijavljivanje na sistem.

Proveriti vreme potrebno za pregled sadržaja kurseva po oblastima i lekcijama.

Proveriti vreme potrebno za obradu popunjene ankete ili testa.

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

**Testiranje opterećenja (Load Testing)**

Proveriti odziv sistema kada mu istovremeno pristupa 700 posetilaca.

Proveriti odziv sistema kada 300 posetilaca istovremeno pristupa pregledu svih kurseva na portalu.

Proveriti odziv sistema kada 100 posetilaca istovremeno pristupa pregledu nekog konkretnog kursa.

Proveriti odziv sistema kada 100 posetilaca istovremeno pristupa pregledu lekcije u okviru kursa.

Proveriti odziv sistema kada 50 studenta u približno istom trenutku sačuva popunjeni test ili anketu.

Proveriti da li sistem može da podrži do 1400 simultanih pristupa korisnika portalu (Arhitekturni projekat, odeljak 11).

**Testiranje kritičnih slučajeva (Stress Testing)**

Proveriti vreme odziva sistema prilikom prvog korišćenja.

Proveriti vreme odziva sistema kada 200 posetilaca istovremeno pristupa pregledu svih kurseva.

**Testiranje volumena (Volume Testing)**

Proveriti odziv sistema kada je 90% kapaciteta diska na serveru popunjeno.

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

Proveriti mogućnost prijavljivanja na sistem sa udaljenog računara.

Proveriti prava pristupa korisnika koji pripadaju različitim grupama (Posetilac, Student, Profesor,Admin ).

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

Proveriti da li je portal Adiutor dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 2%. (Vizija sistema, odeljak 9; Arhitekturni projekat, odeljak 12)

Proveriti da li srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 500 sati (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 IV sa 512MB operativne memorije, kao i na telefonima sledeće konfiguracije ili bolje:

* Css resolution 240x320
* Pixel resolution 240x320
* Operating system Android 2.3.5, iOS, Windows Phone

(Specifikacija zahteva, odeljak 7.6).

Proveriti da li klijentski deo portala Adiutor može da se koristi pomoću Web čitača: (Specifikacija sistema, odeljak 7.6)

* + - Google Chrome 30+
    - FireFox (Mozilla) 20+
    - Opera 20+

**Testiranje instalacije (Installation Testing)**

Proveriti instalaciju sistema na serveru.

# 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 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.

## Tipovi testiranja (Testing Types)

### 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(phpMyAdmin) da bi se direktno uneli i promenili podaci u bazi podataka. 2. Procesi moraju da se ručno pokrenu. 3. Koristiti bazu sa minimalnim brojem podataka, kako bi se povećala preglednost i lakše uočili nepravilnosti. |

### Testiranje sistema (System Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi korektno kretanje kroz portal, 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. |
| Kriterijum završetka: | 1. Svi planirani testovi su izvršeni. 2. Svi identifikovane greške su adekvatno obrađene. |
| Napomene: | 1. Potrebno je obezbediti pristup serveru (npr. web hostingu) na kome će biti instaliran sistem. |

### 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. Utvrditi da li se očekivani rezultati javljaju kada se radi sa tačnim podacima. 4. Utvrditi da li se predviđene poruke o grešci ili upozorenja javljaju kada se radi sa netačnim podacima. 5. 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. |

### Testiranje korisničkog interfejsa (User Interface Testing)

|  |  |
| --- | --- |
| Cilj testiranja: | Potvrditi sledeće:   1. Navigacija kroz portal korektno odslikava poslovne funkcije i zahteve, uključujući od kretanje između stranica, polja i metode pristupa (prelazak tabom, kretanje miša, prečice sa tastature) |
| Tehnike: | 1. Kreirati/modifikovati testove za svaku stranicu da bi se potvrdila odgovarajuća navigacija i stanja objekata za svaku stranicu i objekat portala. |
| Kriterijum završetka: | Svaka stranica treba da se potvrdi da je konzistentan sa referentnom verzijom ili prihvaćenim standardima. |
| Napomene: |  |

### 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. |

### 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. |

### Testiranje kritičnih slučajeva(Stress Testing)

|  |  |
| --- | --- |
| Cilj testa: | Proveriti da li sistem funkcioniše korektno i bez grešaka u sledećim kritičnim slučajevima:   1. maksimalan broj klijenata (stvarnih ili simuliranih) koji rade na sistemu 2. više korisnika koji zahtevaju istu transakciju nad istim podacima. 3. malo ili nimalo memorije na serveru     Napmena: Cilj testiranja kritičnih slučajeva može biti i određivanje uslova pod kojima sistem nije u mogućnosti da nastavi da radi korektno. |
| Tehnike: | 1. Koristiti testove koji su napravljeni za testiranje performansi. 2. Da bi se obavilo testiranje pod ograničenjima, testovi treba da se obaljaju sa jednog računara, a veličinu RAM i spoljne memorije treba smanjiti. 3. 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. |
| Napomene: | 1. Testiranje mreže može zahtevati korišćenje alata koji omogućavaju povećanje protoka podataka na mreži. 2. Veličina spoljne memorije treba da bude privremeno smanjena kako bi smanjila prostor koji može da koristi baza podataka. 3. Testiranje zahteva sinhronizaciju istovremenog pristupa klijenata istim slogovima. |

### Testiranje volumena (Volume Testing)

|  |  |
| --- | --- |
| Cilj testa: | Potvrditi da sistem funkcioniše uspešno u sledećim slučajevima:   1. maksimalan () broj klijenata koji su povezani (simulirani) i koji izvršavaju iste poslovne funkcije koje čine najgori slučaj za duži period vremena. 2. Maksimalna veličina baze podataka je dostignuta i više transakcija koje izvršavaju upite i pribavljaju podatke istovremeno. |
| Tehnike: | 1. Koristiti testove koji su napravljeni za testiranje performansi. 2. 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. 3. 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: | 1. Odabrati period vremena koji će biti prihvatljiv za testiranje uslova velikog volumena. |

### 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: |  |

### 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. Prekid komunikacije kroz mrežnu 2. Nekompletni ciklus (prekid u procesu filtriranja podataka, prekid u procesu sinhronizacije podataka) 3. Netačni pokazivači (ključevi) u bazi podataka 4. 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 na mreži: simulirati ili izazvati gubitak komunikacije kroz mrežu (fizički isključiti komunikacioni kabl)   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 phpMyAdmin alata). 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. 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. |

### 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 se aplikacije tipično koriste na klijentu? 2. Koje podatke aplikacija koristi (koji je obim tih podataka) 3. Ceo sistem, mrežni serveri, baze podataka i slično treba biti opisano ka deo ovog testa. |

### 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 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: | 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? |

## Alati

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Faza** | **Alat** | **Proizvođač/ matični proiz.** | **Verzija** |
| Test Management |  |  | TBD |
| Test Design |  |  | 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 | phpMyAdmin | MySQL | TBD |

# 

# Resursi

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

## 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 | Vladimir Dinić | Obezbeđuje nadgledanje upravljanjem procesa  Odgovornosti:   1. Obezbediti tehničku direkciju 2. Prikupljanje odgovarajućih resursa 3. Upravljanje izveštavanjem |
| Projektant testova | Nenad Dimitrijević | 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 | Nenad Dimitrijević  Dušan Cvetković | Izvršiti testiranje  Odgovornosti:   1. Izvršiti testiranje 2. Zabeležiti rezultate 3. Popraviti greške 4. Opisati greške |
| Administrator testiranja sistema | Vladimir Dinić | 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 | Vladimir Dinić, Nenad Dimitrijević | Pripremiti podatke i bazu podataka za testiranje i utvrditi da njihovo stanje odgovara pretpostavkama.  Odgovornosti:   1. Administrirati podatke (bazu podataka) za testiranje |
| Projektant | Dušan Cvetković | 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 | Nenad Dimitrijević,  Dušan Cvetković | Implementirati testove i rasporediti ih u odgovarajuće klase i pakete testova.  Odgovornosti:   1. Kreirati klase i pakete testova implementirane u test modelu. |

## Sistem

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

|  |  |
| --- | --- |
| **Sistemski resursi** | |
| **Resurs** | **Ime / Tip / Serijski broj** |
| Server | Adiutor.cf,  poseban wamp server na računaru svakog od članova tima |
| Klijentski Test Računari |  |
| —3 udaljena računara  (sa internet pristupom) | Vladimirov računar  Nenadov računar  Dušanov računar |
| Test repozitorijum |  |
| —Server | Adiutor.cf |
| Računari za razvoj testova – 3 | Vladimirov računar  Nenadov računar  Dušanov računar |
| Simulator opterećenja | Posebne skripte smeštene na:   * Adiutor.cf, * poseban wamp server na računaru svakog od članova tima |

# Kontrolne tačke testiranja

Aktivnosti testiranja zavise od iteracija u razvoju softvera. Ciklus razvoja softvera Adiutor 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 | 04. 05. 2016 | 11. 05. 2016 |

# Izlazni produkti testiranja—treba mi ono odozgo da bi ovo napravio

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokument** | **Vlasnik** | **Ocena / Raspodela** | **Rok** |
| Plan Testiranja | Vladimir Dinić | Rukovodioci testiranja | 07. 05. 2016 |
| Test okruženje | Vladimir Dinić | - | 07. 05. 2016 |
| Model testiranja | Nenad Dimitrijević | Interna ocena | 07. 05. 2016 |
| Skup test podataka | Nenad Dimitrijević | Interna ocena | 08. 05. 2016 |
| Test procedure | Dušan Cvetković | Interna ocena | 08. 05. 2016 |
| Test skripte | Dušan Cvetković | - | 08. 05. 2016 |
| Test podrška | Vladimir Dinić | - | 08. 05. 2016 |
| Izveštaj o uočenim greškama | Nenad Dimitrijević, Dušan Cvetković | Rukovodioci testiranja | 11. 05. 2016 |
| Rezultati testa | Dušan Cvetković | Rukovodilac testiranja | 11. 05. 2016 |
| Izveštaj o evaluaciji testa | Vladimir Dinić | Rukovodilac testiranja | 11. 05. 2016 |

## Model testiranja

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

# 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 |