*LoveCooking*

Portal za ljubitelje kuvanja

Plan testiranja

Verzija 1.0

Istorija revizija

| **Datum** | **Verzija** | **Opis** | **Autor** |
| --- | --- | --- | --- |
| 14.05.2021. | 1.0 | Inicijalna verzija | Andrija Cenić 18042  Anastasija Stajković 17933  Anđela Cvetković 18034 |

Sadržaj

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

[**2.**](#_30j0zll) **Opseg 4**

[**3.**](#_1fob9te) **Reference 4**

[**4.**](#_3znysh7) **Zahtevi za testiranje 4**

[**5.**](#_2et92p0) **Strategije testiranja (Test Strategy) 8**

[5.1](#_tyjcwt) Tipovi testiranja (Testing Types) 8

[*5.1.1*](#_3dy6vkm) *Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing) 8*

[*5.1.2*](#_1t3h5sf) *Testiranje sistema (System Testing) 8*

[*5.1.3*](#_2s8eyo1) *Testiranje poslovnog ciklusa (Business Cycle Testing) 9*

[*5.1.4*](#_17dp8vu) *Testiranje korisničkog interfejsa (User Interface Testing) 10*

[*5.1.5*](#_26in1rg) *Testiranje performansi (Performance Testing) 10*

[*5.1.6*](#_lnxbz9) *Testiranje opterećenja (Load Testing) 11*

[*5.1.7*](#_35nkun2) *Testiranje kritičnih slučajeva**(Stress Testing) 11*

[*5.1.8*](#_44sinio) *Testiranje volumena (Volume Testing) 12*

[*5.1.9*](#_2jxsxqh) *Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing) 13*

[*5.1.10*](#_z337ya) *Testiranje otkaza i oporavka (Failover / Recovery Testing) 13*

[*5.1.11*](#_1y810tw) *Testiranje konfiguracije (Configuration Testing) 15*

[*5.1.12*](#_4i7ojhp) *Testiranje instalacije (Installation Testing) 15*

[5.2](#_1ci93xb) Alati 16

[**6.**](#_qsh70q) **Resursi 16**

[6.1](#_1pxezwc) Radnici 16

[6.2](#_49x2ik5) Sistem 18

[**7.**](#_2p2csry) **Kontrolne tačke testiranja 18**

[**8.**](#_147n2zr) **Izlazni produkti testiranja 18**

[8.1](#_3o7alnk) Model testiranja 19

[**9.**](#_23ckvvd) **Projektni zadaci 19**

Plan Testiranja

# Ciljevi

Ovaj dokument sadrži opis plana testiranja sistema ***LoveCooking***. 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 ***LoveCooking*** Web 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 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 ***LoveCooking***,
2. Specifikacija zahteva sistema ***LoveCooking*** i
3. Arhitekturni projekat ***LoveCooking***.

# 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 ***LoveCooking***.

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 naslovne strane portala.

Proveriti slučaj korišćenja Pregled recepata.

Proveriti slučaj korišćenja Pregled takmičenja.

Proveriti slučaj korišćenja Pregled menija.

Proveriti slučaj korišćenja Pregled profila korisnika.

Proveriti slučaj korišćenja Registracija korisnika.

Proveriti slučaj korišćenja Kreiranje naloga.

Proveriti slučaj korišćenja Prijavljivanje korisnika.

Proveriti slučaj korišćenja Kreiranje recepta.

Proveriti slučaj korišćenja Dodavanje novog sastojka.

Proveriti slučaj korišćenja Predlog novog sastojka.

Proveriti slučaj korišćenja Kreiranje menija.

Proveriti slučaj korišćenja Komentarisanje recepta.

Proveriti slučaj korišćenja Ocenjivanje recepta.

Proveriti slučaj korišćenja Kreiranje challenge-a.

Proveriti slučaj korišćenja Glasanje na challenge-u.

Proveriti slučaj korišćenja Prijava na challenge.

Proveriti slučaj korišćenja Sortiranje recepta na osnovu cene.

Proveriti slučaj korišćenja Sortiranje recepta na osnovu kalorija.

Proveriti slučaj korišćenja Sortiranje recepta na osnovu ocene.

Proveriti slučaj korišćenja Sortiranje recepta na osnovu vremena pripreme.

Proveriti slučaj korišćenja Filtriranje recepta na osnovu naziva.

Proveriti slučaj korišćenja Filtriranje recepta na osnovu vremena pripreme.

Proveriti slučaj korišćenja Filtriranje recepta na osnovu kalorijskog sadržaja.

Proveriti slučaj korišćenja Filtriranje recepta na osnovu tipa jela.

Proveriti slučaj korišćenja Filtriranje recepta na osnovu autora.

Proveriti slučaj korišćenja Pretraživanje recepata.

Proveriti slučaj korišćenja Pretraživanje menija.

Proveriti slučaj korišćenja Pretraživanje korisnika.

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

Serverska komponenta sistema treba da funkcioniše na računaru sa R5-4600H i 8GB RAM memorije.

Klijentska komponenta sistema treba da funkcioniše na računaru sa Pentium IV procesorom i 512 MB RAM memorije (Specifikacija zahteva, odeljak 7.6).

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

Proveriti posledice operacije dodavanja novog Korisnika.

Proveriti posledice operacije dodavanja novog Recepta.

Proveriti posledice operacije dodavanja novog Menija.

Proveriti posledice operacije dodavanja novog Takmičenja.

Proveriti posledice operacije dodavanja novog Komentara.

Proveriti posledice operacije dodavanja novog Sastojka.

Proveriti posledice operacije ažuriranja profila Korisnik.

Proveriti posledice operacije brisanja profila Korisnik.

Proveriti posledice operacije brisanja Recepta.

Proveriti posledice operacije brisanja Menija.

Proveriti posledice operacije brisanja Takmičenja.

Proveriti posledice operacije brisanja Komentara.

Proveriti posledice operacije brisanja Sastojka.

Proveriti posledice operacije definisanja novog Recepta.

Proveriti posledice operacije definisanja novog Takmičenja.

Proveriti posledice operacije definisanja novog Menija.

Proveriti posledice operacije definisanja novog Komentara.

Proveriti posledice operacije definisanja novog Sastojka.

Proveriti posledice operacije definisanja novog Korisnika.

**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 fleksibilani da li je moguće korišćene na rayličitim uređajima.

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 korisnika.

Proveriti vreme potrebno za dodavanje novog menija.

Proveriti vreme potrebno za dodavanje novog sastojka.

Proveriti vreme potrebno za dodavanje novog recepta.

Proveriti vreme potrebno za dodavanje novog takmičenja.

Proveriti vreme potrebno za dodavanje novog komentara.

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 200 posetilaca.

Proveriti odziv sistema kada 100 posetilaca istovremeno pristupa stranici za pregled predavača.

Proveriti odziv sistema kada 10 posetilaca istovremeno pristupa stranici profila predavača.

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 za pregled poečetne strane portala.

Proveriti vreme odziva sistema kada 100 posetilaca istovremeno pokušava da se registruje.

**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 (Posetilac portala - Gost, Prijavljeni korisnik, Administrator).

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

Proveriti da li je ***LoveCooking*** Web portal dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 15%. (Specifikacija zahteva 7.3, Arhitekturni projekat odeljak 11).

Proveriti da li srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 96 sati (Arhitekturni projekat, odeljak 11).

**Testiranje konfiguracije (Configuration Testing)**

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

Klijentski deo sistema treba da ima mogućnost izvršavanja na bilo kom PC računaru sa minimum Pentium IV procesorom i 512 MB RAM memorije (Specifikacija zahteva, odeljak 7.6).

Klijentski deo ***LoveCooking*** portala će biti optimizovan za sledeće Web čitače: Chrome( browseri bazirani na choromium), Mozila Firefox, Safari, Opera... (Većina modernih browser-a) (Specifikacija sistema, odeljak 7.6).

**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 koji način 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: | * 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 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.Procesi podrazumevaju unos podataka, brisanje podataka ili izmenu podataka. |
| Napomene: | * Testiranje zahteva korišćenje okruženja za rad sa konkretnim DBMS-om da bi se direktno uneli i promenili podaci u bazi podataka. * Procesi moraju da se ručno pokrenu. * Koristiti bazu sa malim ili minimalnim brojem podataka, kako bi se povećala preglednost i lakše uočili neprihvatljivi slučajevi. |

### Testiranje sistema (System Testing)

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

### 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: | * Testiranje simulira poslovne cikluse koji obavljaju sledeće: * Testovi korišćeni za testiranje funkcija aplikacija treba izmeniti/proširiti tako da uvećavaju 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 korišćenjem korektnih i nekorektnih podataka. * Proveriti da li se sve funkcije, koje se pojavljuju periodično, izvršavaju/pokreću u odgovarajućim vremenskim trenucima. * Utvrditi da li se očekivani rezultati javljaju kada se radi sa tačnim podacima. * Utvrditi da li se predviđene poruke o grešci ili upozorenja javljaju kada se radi sa netačnim podacima. * Potvrditi da su poslovna pravila pravilno primenjena. |
| Kriterijum završetka: | * Svi planirani testovi su izvršeni. * Sve identifikovane greške su adekvatno obrađene. |
| Napomene: | * Određivanje datuma i perioda može zahtevati dodatne aktivnosti. * Poslovni model bi trebalo da identifikuje odgovarajuće zahteve i procedure koje treba testirati. |

### Testiranje korisničkog interfejsa (User Interface Testing)

| Cilj testiranja: | Potvrditi sledeće:   * Navigacija kroz aplikaciju korektno oslikava poslovne funkcije i zahteve, uključujući kretanje između prozora, polja i metode pristupa (prelazak tabom, kretanje miša) * 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: |  |

### Testiranje performansi (Performance Testing)

| Cilj testiranja: | Potvrditi vreme odziva sistema za napravljene transakcije ili poslovne funkcije za sledeća dva slučaja:  - prosečno procenjeno opterećenje  - maksimalno procenjeno opterećenje |
| --- | --- |
| Tehnike: | * Koristiti test procedure napravljene za testiranje poslovnog modela (testiranje sistema) * Modifikovati podatke tako da se uveća broj transakcija i modifikovati skripte da se poveća broj izvršenja transakcija. * Skripte treba da se izvršavaju na jednom računaru (jedan korisnik, jedna transakcija) ili ponoviti sa više klijenata (virtuelinih ili stvarnih). |
| Kriterijum završetka: | * Jedna transakcija / jedan korisnik: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme. * Više transakcija / više korisnika: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme. |
| Napomena: | * 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 MicrosoftSQL poziva. * 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. * Koristiti više ljudi koji će sa različitih računara pokretati test skripte da bi se dostiglo odgovarajuće opterećenje. * Testiranje performansi treba da se obavi na za to spremljenom računaru u predviđeno vreme. Time se omogućava potpuna kontrola i precizno merenje. * 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: | * 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 završetka: | * Više transakcija / više korisnika: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme. |
| Napomene: | * Testiranje performansi treba da se obavi na za to pripremljenom računaru 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 u sistemu.. |

### 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.   Napomena: Cilj testiranja kritičnih slučajeva može biti i određivanje uslova pod kojima sistem nije u mogućnosti da nastavi da nastavi sa korektnim radom.. |
| --- | --- |
| 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 najvećom 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 izvan predviđenih granica. |
| Napomene: | * Testiranje mreže može zahtevati korišćenje alata koji omogućavaju povećanje protoka podataka na mreži. * Veličina eksterne memorije treba da bude privremeno smanjena kako bi smanjila prostor koji može da koristi baza podataka. * Testiranje zahteva sinhronizaciju istovremenog pristupa klijenata istim slogovima. |

### 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 najvećom 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. |

### 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: | * Sigurnost funkcija / podataka: Identifikovati funkcije/podatke kojima ima pravo pristupa odgovarajući tip korisnika. * Napraviti testove za svaki tip korisnika i potvrditi dozvolu da izvršava transakcije koje su specifične za taj tip korisnika. * 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 oporavka (ručni ili automatski) korektno vraća bazu podataka, aplikaciju i sistem u očekivano stanje. Sledeći tipovi uslova treba da budu uključeni u testiranje:   * Isključenje napajanja na klijentu * Isključenje napajanja na serveru * Prekid komunikacije kroz mrežnu * Prekid komunikacije ili gubitak podataka usled greške rada hard diska * Nekompletni ciklus (prekid u procesu filtriranja podataka, prekid u procesu sinhronizacije podataka) * Netačni pokazivači (ključevi) u bazi podataka * Netačni podaci 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:   * Prekid napajanja na klijentu * Prekid napajanja na serveru * Prekid na mreži: simulirati ili izazvati gubitak komunikacije kroz mrežu (fizički isključiti komunikacioni kabl ili isključiti mrežni server ili ruter) * 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: | * Testiranje oporavka zahteva operacije koje nekada nisu izvodljive ili ih nije poželjno obavljati (simulacija prekida komunikacije ili napajanja). Alternativa tome su softverski alati za dijagnostiku. * Resursi sistema (računara), baze podataka i mrežne grupe se koriste. * 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: | * 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 grešaka. |
| Napomene: | * Koje su aplikacije dostupne i mogu se koristiti na klijentu? * Koje se aplikacije tipično koriste? * Koje podatke aplikacija koristi (koji je obim tih podataka) * 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:   * Nova instalacija na novom računaru, nikad instalirana aplikacija. * Ažurirati računar prethodno instaliran sa istom verzijom aplikacije * Ažurirati računar prethodno instaliran sa prethodnom verzijom aplikacije |
| --- | --- |
| Tehnike: | * Ručno ili definisanjem automatskih skripti proveriti uslove na ciljnoj mašini (nikad instalirana aplikacija, ista verzija, nova verzija). * Pokrenuti i instalirati aplikaciju. * 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 | Visual Studio Code+Mocha | Microsoft | TBD |
| Test Design | Visual Studio Code+Mocha | Microsoft | TBD |
| Defect Tracking |  |  | TBD |
| Functional Testing |  |  | TBD |
| Performance Testing |  |  | TBD |
| Test Coverage Monitor or Profiler |  |  | TBD |
| Other Test Tools |  |  | TBD |
| Project Management | Google Docs  Word  Trello | Google  Microsoft  Atlassian | TBD |
| DBMS tools | Microsoft SQL Server - localdb - offline testing,  Microsoft Azure SQL Server - online database | Microsoft  Microsoft | TBD  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 | Andrija Cenić,  Anastasija Stajković | Obezbeđuje nadgledanje upravljanjem procesa  Odgovornosti:   * Obezbediti tehničku direkciju * Prikupljanje odgovarajućih resursa * Upravljanje izveštavanjem |
| Projektant testova | Anastasija Stajković  Anđela Cvetković | Identifikuje, određuje prioritete i implementira slučajeve testiranja  Odgovornosti:   * Napraviti plan testiranja * Napraviti model testiranja * Proceniti efektivnost napora uloženog u testiranje. |
| Tester sistema | Anastasija Stajković  Anđela Cvetković | Izvršiti testiranje  Odgovornosti:   * Izvršiti testiranje * Zabeležiti rezultate * Popraviti greške * Opisati greške |
| Administrator testiranja sistema | Andrija Cenić | Pripremiti okruženje za testiranje i utvrditi da ono odgovara pretpostavkama.  Odgovornosti:   * Administrirati upravljanje testiranjem * Upravljati pristupom radnika resursima na kojima se obavlja testiranje. |
| Administrator baze podataka | Anđela Cvetković | Pripremiti podatke i bazu podataka za testiranje i utvrditi da njihovo stanje odgovara pretpostavkama.  Odgovornosti:   * Administrirati podatke (bazu podataka) za testiranje |
| Projektant | Andrija Cenić  Anastasija Stajković  Anđela Cvetković | Identifikovati i definisati operacije, atribute i veze između različitih klasa testova  Odgovornosti:   * Identifikovati i definisati klase testova * Identifikovati i definisati pakete testova |
| Implementacija | Andrija Cenić  Anđela Cvetković | Implementirati testove i rasporediti ih u odgovarajuće klase i pakete testova.  Odgovornosti:   * Kreirati klase i pakete testova implementirane u test modelu. |

## Sistem

U sledećoj tabeli su navedeni sistemski resursi koji će se koristiti za testiranje ***Professorry*** 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 |

# Kontrolne tačke testiranja

Aktivnosti testiranja zavise od iteracija u razvoju softvera. Ciklus razvoja sistema ***Professorry*** 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 | Maj 10 | Maj 14 |

# Izlazni produkti testiranja

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

| **Dokument** | **Vlasnik** | **Ocena / Raspodela** | **Due Date** |
| --- | --- | --- | --- |
| Plan Testiranja | Andrija Cenić | Rukovodilac testiranja | Maj 10 |
| Test okruženje | Anastasija Stajković | - | Maj 10 |
| Model testiranja | Anđela Cvetković | Interna ocena | Maj 11 |
| Skup test podataka | Andrija Cenić | Interna ocena | Maj 11 |
| Test procedure | Andrija Cenić  Anastasija Stajković | Interna ocena | Maj 12 |
| Test skripte | Andrija Cenić | - | Maj 12 |
| Test podrška | Andrija Cenić  Anđela Cvetković | - | Maj 13 |
| Izveštaj o uočenim greškama | Anastasija Stajković | Rukovodilac testiranja | Maj 13 |
| Rezultati testa | Anđela Cvetković | Rukovodilac testiranja | Maj 14 |
| Izveštaj o evaluaciji testa | Anastasija Stajković  Anđela Cvetković | Rukovodioci testiranja | Maj 14 |

## 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 |