
Lambda

TestIT
Plan testiranja

Verzija 1.0

Lambda

Istorija revizija

Datum	Verzija	Opis	Autor
30.4.2019	1.0	Inicijalna verzija	Tim Lambda
4.5.2019	1.1	Ispravljene pravopisne greske	Tim Lambda

Sadržaj

Ciljevi	4
Opseg	4
Reference	4
Zahtevi za testiranje	4
Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing)	5
Testiranje sistema – funkcionalno testiranje (System Testing – functional testing)	5
Testiranje poslovnog ciklusa (Business Cycle Testing)	5
Testiranje korisničkog interfejsa (User Interface Testing)	6
Testiranje performansi (Performance Testing)	6
Testiranje opterećenja (Load Testing)	6
Testiranje kritičnih slučajeva (Stress Testing)	6
Testiranje volumena (Volume Testing)	6
Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing)	6
Testiranje otkaza i oporavka (Failover / Recovery Testing)	7
Testiranje konfiguracije (Configuration Testing)	7
Testiranje instalacije (Installation Testing)	7
Strategije testiranja (Test Strategy)	7
Tipovi testiranja (Testing Types)	7
Testiranje podataka i integriteta baze podataka (Data and Database Integrity Testing)	7
Testiranje sistema (System Testing)	7
Testiranje poslovnog ciklusa (Business Cycle Testing)	8
Testiranje korisničkog interfejsa (User Interface Testing)	9
Testiranje performansi (Performance Testing)	9
Testiranje opterećenja (Load Testing)	10
Testiranje kritičnih slučajeva (Stress Testing)	10
Testiranje volumena (Volume Testing)	11
Testiranje sigurnosti i kontrole pristupa (Security and Access Control Testing)	12
Testiranje otkaza i oporavka (Failover / Recovery Testing)	12
Testiranje konfiguracije (Configuration Testing)	13
Testiranje instalacije (Installation Testing)	14
Alati	14
Resursi	15
Radnici	15
Sistem	16
Kontrolne tačke testiranja	16
Izlazni produkti testiranja	17
Model testiranja	17
Projektni zadaci	17

Plan Testiranja

1. Ciljevi

Ovaj dokument sadrži opis plana testiranja TestIT aplikacije. 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 testiranju.
- Naveći dokumente koji će opisati rezultate testiranja.

2. Opseg

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

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.

3. Reference

Reference na osnovu kojih će biti napisan Plan testiranja su:

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

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 TestIT.

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 osnovnih podataka o predmetu.

Proveriti slučaj korišćenja Pregled podataka o članovima

Proveriti slučaj korišćenja Pregled registrovanih korisnika.

Proveriti slučaj korišćenja Pregled ličnih podataka.

Proveriti slučaj korišćenja Pregled podataka o kvizu.

Proveriti slučaj korišćenja Pregled komentara.

Proveriti slučaj korišćenja Pregled podataka o takmičenjima.

Proveriti slučaj korišćenja Registracija.

Proveriti slučaj korišćenja Prijavljivanje (*Login*).

Proveriti slučaj korišćenja Prijavljivanje na predmet.

Proveriti slučaj korišćenja Ažuriranje sopstvenih podataka.

Proveriti slučaj korišćenja Dodavanje novih kvizova.

Proveriti slučaj korišćenja Brisanje postojećih kvizova.

Proveriti slučaj korišćenja Izmena postojećih kvizova.

Proveriti slučaj korišćenja Učestvovanje u kviz.

Proveriti slučaj korišćenja Validacija kviza od strane moderatora.

Proveriti slučaj korišćenja Dodavanje pitanja na postojeće kvizove.

Proveriti slučaj korišćenja Izmena postojećih pitanja.

Proveriti slučaj korišćenja Brisanje pitanja.

Proveriti slučaj korišćenja Kreiranje turnira.

Proveriti slučaj korišćenja Učestvovanje na turnir.

Proveriti slučaj korišćenja Dodavanje novog predmeta.

Proveriti slučaj korišćenja Strukturisanje predmeta

Serverska komponenta sistema treba da funkcioniše na PC računaru sa minimum jednojezgarnim procesorom i 1GB RAM memorije i bilo kojim modernim operativnim sistemom.

Klijentska komponenta sistema treba da funkcioniše na bilo kom uređaju koji podržava rad sledećih web čitaca: Internet Explorer 11.0 i noviji, Opera, Firefox, Google Chrome i ostali “moderni” čitači.

Testiranje poslovnog ciklusa (Business Cycle Testing)

Proveriti posledice operacije dodavanja novog predmeta.

Proveriti posledice operacije dodavanja novog profesora.

Proveriti posledice operacije dodavanja novog kviza.

Proveriti posledice operacije dodavanja novog turnira.

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 registraciju na sistem.

Proveriti vreme potrebno za prijavljivanje na sistem.

Proveriti vreme potrebno za dodavanje novog kviza.

Proveriti vreme potrebno za dodavanje kreiranje turnira.

Proveriti da li se podaci u bazi podataka ažuriraju za manje od 5 sekundi.

Proveriti da li se podaci iz baze podataka pribavljaju za manje od 5 sekundi (Specifikacija zahteva, odeljak 7.4).

Testiranje opterećenja (Load Testing)

Proveriti odziv sistema kada mu istovremeno pristupa 20 (100+)¹ posetilaca.

Proveriti odziv sistema kada 20 (100+) posetilaca istovremeno pristupa stranici Liste svih predmeta.

Proveriti odziv sistema kada 20 (50+) posetilaca istovremeno pristupa stranici Liste svih kvizova na jednom predmetu.

Provera odziva sistema kada 20 (50+) korisnika pristupe stranici za odradu istog kviza.

Provera odziva sistema kada 20 (50+) korisnika pristupaju istom turniru.

Testiranje kritičnih slučajeva (Stress Testing)

Proveriti vreme odziv sistema prilikom prvog korišćenja.

Proveriti vreme odziva sistema kada 50+ posetilaca istovremeno pokušava da odradi kviz.

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 udaljenog računara.

Proveriti prava pristupa korisnika koji pripadaju različitim grupama (Posetilac aplikacije, Registrovani korisnik/ Student, Registrovani korisnik/ Profesor, Moderator, Administrator).

Proveriti da li korisnici imaju pristup podacima o drugim korisnicima koji nisu namenjeni za prikaz.

Testiranje otkaza i oporavka (Failover / Recovery Testing)

Proveriti da li je TestIT aplikacija dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupna 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).

¹ planirani opseg je napisan u zagradama, sa ogranicenim resursima mozemo da testiramo samo sa 20 osoba istovremeno

Testiranje konfiguracije (Configuration Testing)

Proveriti da li serverska komponenta TestIT aplikacije može da funkcioniše na PC računaru sa minimum jednojezgarnim procesorom i 1GB RAM memorije i bilo kojim modernim operativnim sistemom.

Proveriti da li klijentski deo TestIT aplikacije može da se koristi pomoću Web čitača: Internet Explorer 11.0 i noviji, Opera, Firefox, Google Chrome (Specifikacija sistema, odeljak 7.6).

Testiranje instalacije (Installation Testing)

Proveriti da li je TestIT aplikacija pravilno instalirana na Azure platformi.

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 na kako će navedeni elementi biti testirani.

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

5.1 Tipovi testiranja (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:	<ul style="list-style-type: none">• 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.
Napomene:	<ul style="list-style-type: none">• Proces i moraju da se ručno pokrenu.• 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:	<ul style="list-style-type: none">• Izvršiti svaki slučaj korišćenja, ispratiti sve tokove definisane slučajevima korišćenja, korišćenjem tačnih i netačnih podataka.• Utvrditi da li se očekivani rezultati javljaju kada se radi sa tačnim podacima.

Kriterijum završetka:	<ul style="list-style-type: none">• Utvrditi da li se predviđene poruke o grešci ili upozorenja javljaju kada se radi sa netačnim podacima.• Potvrditi da su poslovna pravila pravilno primenjena.• Svi planirani testovi su izvršeni.• Svi identifikovane greške su adekvatno obrađene.
Napomene:	<ul style="list-style-type: none">• 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:	<ul style="list-style-type: none">• Testiranje simulira poslovne cikluse koji obavljaju sledeće:<ul style="list-style-type: none">• 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.• Sve funkcije koje su vremenski zavisne treba proveriti korišćenjem tačnih i netačnih podataka.• Proveriti da li se sve funkcije koje se pojavljuju periodično da se izvršavaju/pokreću u odgovarajućem 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:	<ul style="list-style-type: none">• Svi planirani testovi su izvršeni.• Svi identifikovane greške su adekvatno obrađene.
Napomene:	<ul style="list-style-type: none">• Određivanje datuma i perioda može zahtevati dodatne aktivnosti.• 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:
------------------	--------------------

	<ul style="list-style-type: none">• 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:	<ul style="list-style-type: none">• 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: <ul style="list-style-type: none">- procenjeno prosečno opterećenje- procenjeno maksimalno opterećenje
Tehnike:	<ul style="list-style-type: none">• 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:	<ul style="list-style-type: none">• 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:	<ul style="list-style-type: none">• Testiranje performansi sistema uključuje opterećenje sistema u pozadini. Postoji više metoda koji se mogu koristiti da bi se dobili zahtevani uslovi:<ul style="list-style-type: none">• Pozivati transakcije direktno na serveru, obično u formi SQL poziva.• Kreirati virtuelnog korisnika koji će simulirati više klijenata. Može se iskoristiti softver koji će sa udaljenog 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 računar u predviđeno vreme. Time se omogućuje potpuna kontrola i precizno merenje.
- Baza podataka koja se koristi za testiranje performansi treba da ima veličinu baze koja će se realno koristiti.

5.1.6 Testiranje opterećenja (Load Testing)

Cilj testiranja:	Potvrditi vreme odziva sistema za napravljene transakcije i poslovne slučajeve pod uslovima različitog opterećenja.
Tehnike:	<ul style="list-style-type: none">• 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:	<ul style="list-style-type: none">• Više transakcija / više korisnika: Uspešno izvršavanje test skripti bez ikakvih padova i za očekivano vreme.
Napomene:	<ul style="list-style-type: none">• Testiranje performansi treba da se obavi na za to spremljenom računar u predviđeno vreme. Time se omogućuje potpuna kontrola i precizno merenje.• Baza podataka koja se koristi za testiranje performansi treba da ima veličinu baze koja će se realno koristiti.

5.1.7 Testiranje kritičnih slučajeva (Stress Testing)

Cilj testa:	<p>Proveriti da li sistem funkcioniše korektno i bez grešaka u sledećim kritičnim slučajevima:</p> <ul style="list-style-type: none">• 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.
	<p>Napomena: Cilj testiranja kritičnih slučajeva može biti i određivanje uslova pod kojima sistem nije u mogućnosti da nastavi da radi korektno.</p>
Tehnike:	<ul style="list-style-type: none">• Koristiti testove koji su napravljeni za testiranje performansi.• Da bi se obavilo testiranje pod ograničenjima, testovi treba da se obalaju sa jednog računara, a veličinu RAM i spoljne memorije treba smanjiti.

	<ul style="list-style-type: none">• 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:	<ul style="list-style-type: none">• 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: <ul style="list-style-type: none">• 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:	<ul style="list-style-type: none">• 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:	<ul style="list-style-type: none">• 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:	<p>Sigurnost funkcija/podataka: Potvrditi da korisnik može da koristi samo one funkcije/ podatke koje su predviđene za tip korisnika kome pripada.</p> <p>Sigurnost sistema: Potvrditi da samo oni korisnici kojima je dozvoljeno da pristupe sistemu mogu da pristupe sistemu.</p>
Tehnike:	<ul style="list-style-type: none">• 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:	<p>Za svaki tip korisnika i odgovarajuće funkcije/podatke dozvoljene za korišćenje, sve transakcije funkcionišu na očekivani način, definisane u testovima sistema.</p>
Napomene:	

5.1.10 Testiranje otkaza i oporavka (Failover / Recovery Testing)

Cilj testiranja:	<p>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:</p> <ul style="list-style-type: none">• 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 ili podatak u bazi podataka
Tehnike:	<p>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 sprovedu (simulirati), individualno:</p> <ul style="list-style-type: none">• 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 izvodive ili h 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.

5.1.11 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.

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: <ul style="list-style-type: none">• 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:	<ul style="list-style-type: none">• Ručno ili definisanjem automatskih skripti proveriti uslove na ciljnoj mašini (nikad instalirana aplikacija, ista verzija, nova verzija).• Pokrenuti i instalirati aplikaciju.• Korišćenjem predefinisanih podskupa testova funkcionalnih testova, proveriti rad aplikacije.
Kriterijum završetka:	Aplikacija treba da se izvršavaju uspešno i bez grešaka.
Napomene:	<ul style="list-style-type: none">• 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			
Functional Testing	MSTest Unit tests	Microsoft	2017
Performance Testing	Azure performance metrics	Microsoft	
Test Coverage Monitor or Profiler			
Other Test Tools			
Project Management	Project Word	Microsoft Microsoft	2007

	Excel	Microsoft	
DBMS tools	MySQL	MySQL	5.7.9

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		
Uloga radnika Uloga radnika	Radnici (volonteri za testiranje aplikacije)	Posebne preporuke / komentari
Rukovodilac testiranja	Mihajlo Veljković Mladen Stanković Ilija Milosević Stefan Bogosavljević	Obezbeđuje nadgledanje upravljanjem procesa Odgovornosti: <ul style="list-style-type: none"> • Obezbediti tehničku direkciju • Prikupljanje odgovarajućih resursa • Upravljanje izveštavanjem
Projektant testova	Mihajlo Veljković Mladen Stanković Ilija Milosević Stefan Bogosavljević	Identifikuje, određuje prioritete i implementira slučajeve testiranja Odgovornosti: <ul style="list-style-type: none"> • Napraviti plan testiranja • Napraviti model testiranja • Proceniti efektivnost napora uloženog u testiranje.
Tester sistema	Svi volonteri koji žele da testiraju TestIT sistem	Izvršiti testiranje Odgovornosti: <ul style="list-style-type: none"> • Izvršiti testiranje • Zabeležiti rezultate • Popraviti greške • Opisati greške
Administrator testiranja sistema	Mihajlo Veljković	Pripremiti okruženje za testiranje i utvrditi da ono odgovara pretpostavkama. Odgovornosti: <ul style="list-style-type: none"> • Administrirati upravljanje testiranjem • Upravlјati pristupom radnika resursima na kojima se obavlja testiranje.
Administrator baze podataka	Stefan Bogosavljević	Pripremiti podatke i bazu podataka za testiranje i utvrditi da njihovo stanje odgovara pretpostavkama.

		Odgovornosti: <ul style="list-style-type: none"> Administrirati podatke (bazu podataka) za testiranje
Projektant	Mihajlo Veljković Mladen Stanković	Identifikovati i definisati operacije, atribute i veze između različitih klasa testova Odgovornosti: <ul style="list-style-type: none"> Identifikovati i definisati klase testova Identifikovati i definisati pakete testova
Implementacija	Ilija Milosević	Implementirati testove i rasporediti ih u odgovarajuće klase i pakete testova. Odgovornosti: <ul style="list-style-type: none"> 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 TestIT sistema.

Sistemski resursi	
Resurs	Ime / Tip / Serijski broj
Server	Azure instanca: https://test-it.azurewebsites.net/
Klijentski Test Računari 5 računara (sa internet pristupom) + uređaji dobrovoljaca	mešavina desktop/laptop računara i mobilnih uređaja
Računari za razvoj testova - 4	2 laptop i 2 desktop računara

7. Kontrolne tačke testiranja

Aktivnosti testiranja zavise od iteracija u razvoju softvera. Ciklus razvoja TestIT aplikacije biće odraden 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 05	Maj 14

8. Izlazni produkti testiranja

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

Dokument	Vlasnik	Ocena / Raspodela	Rok
Plan Testiranja	Lambda Tim	Rukovodioci testiranja	Maj 07
Test okruženje	Mihajlo	-	Maj 07
Model testiranja	Stefan	Interna ocena	Maj 07
Skup test podataka	Mladen	Interna ocena	Maj 08
Test procedure	Ilija	Interna ocena	Maj 08
Test skripte	Ilija	-	Maj 08
Test podrška	Mladen	-	Maj 08
Izveštaj o uočenim greškama	Stefan	Rukovodioci testiranja	Maj 11
Rezultati testa	Mihajlo	Rukovodilac testiranja	Maj 11
Izveštaj o evaluaciji testa	Mihajlo	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 sledećem odeljku 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 specifične 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