

Plan testiranja **Raspo**red

Tim: *La Plávusha*

Maj 2016

Pregled izmena

Verzija 1.0 (24. april 2016)

Inicijalna verzija dokumenta.

Sadržaj

1	Cilj dokumenta	4
2	Opseg dokumenta	5
3	Reference	6
4	Zahtevi za testiranje	7
4.1	Testiranje podataka i integriteta baze podataka	7
4.2	Testiranje sistema – funkcionalno testiranje	7
4.3	Testiranje poslovnog ciklusa	8
4.4	Testiranje korisničkog interfejsa	9
4.5	Testiranje performansi	9
4.6	Testiranje opterećenja	9
4.7	Testiranje kritičnih slučajeva	10
4.8	Testiranje sigurnosti i kontrole pristupa	10
4.9	Testiranje otkaza i oporavka	10
4.10	Testiranje konfiguracije	10
4.11	Testiranje instalacije	11
5	Strategije testiranja	12
5.1	Tipovi testiranja	12
5.1.1	Testiranje podataka i integriteta baze podataka	12
5.1.2	Testiranje sistema	13
5.1.3	Testiranje poslovnog ciklusa	13
5.1.4	Testiranje korisničkog interfejsa	14
5.1.5	Testiranje performansi	14
5.1.6	Testiranje opterećenja	15
5.1.7	Testiranje kritičnih slučajeva	15
5.1.8	Testiranje volumena	16
5.1.9	Testiranje sigurnosti i kontrole pristupa	17
5.1.10	Testiranje otkaza i oporavka	17
5.1.11	Testiranje konfiguracije	19
5.2	Alati	19

6	Resursi	21
6.1	Radnici	21
6.2	Sistem	21
6.3	Kontrolne tačke testiranja	21
6.4	Izlazni produkti testiranja	21
	6.4.1 Model testiranja	24
6.5	Projektni zadaci	24

Glava 1

Cilj dokumenta

Ovaj dokument sadrži opis plana testiranja sistema **Raspored**. 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.
- Navesti dokumente koji će opisati rezultate testiranja.

Glava 2

Opseg dokumenta

Plan testiranja se odnosi na integraciju sistema i proveru izvršne verzije aplikacije **Raspored**. Plan testiranja ne opisuje testiranje pojedinačnih komponenti sistema, jer se pretpostavlja 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.

Glava 3

Reference

- [1] **Raspo**red – *Predlog projekta*, V1.0.1; 2016, *La Plávusha*.
- [2] **Raspo**red – *Plan realizacije*, V1.0.1; 2016, *La Plávusha*.
- [3] **Raspo**red – *Planirani raspored aktivnosti na projektu*, V1.0; 2016, *La Plávusha*.
- [4] **Raspo**red – *Vizija sistema*, V1.0.1; 2016, *La Plávusha*.
- [5] **Raspo**red – *Specifikacija zahteva*, V1.1; 2016, *La Plávusha*.
- [6] **Raspo**red – *Arhitekturni projekat*, V1.0; 2016, *La Plávusha*.

Glava 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 Specifikacija testova.

4.1 Testiranje podataka i integriteta baze podataka

- Proveriti mogućnost pristupa bazi podataka **Raspored**.
- Proveriti mogućnost istovremenog pristupa istim podacima radi čitanja.
- Proveriti zaključivanje podataka tokom pristupa bazi ažuriranja.
- Proveriti tačnost podataka koji se pribavljaju iz baze.

4.2 Testiranje sistema – funkcionalno testiranje

- Proveriti slučaj korišćenja *Prijavljivanje na sistem*.
- Proveriti slučaj korišćenja *Pregled rasporeda (globalnog, zvaničnog i ličnog)*.
- Proveriti slučaj korišćenja *Dodavanje časa u lični raspored*.
- Proveriti slučaj korišćenja *Dodavanje aktivnosti u lični raspored*.
- Proveriti slučaj korišćenja *Sakrivanje aktivnosti iz ličnog rasporeda*.
- Proveriti slučaj korišćenja *Pregled informacija vezanih za konkretan čas*.

- Proveriti slučaj korišćenja *Dodavanje taskova*.
- Proveriti slučaj korišćenja *Pretraživanje oglasne table*.
- Proveriti slučaj korišćenja *Dodavanje oglasa na oglasnu tablu*.
- Proveriti slučaj korišćenja *Pregled rasporeda*.
- Proveriti slučaj korišćenja *Pregled grupa*.
- Proveriti slučaj korišćenja *Pregled studenata jedne grupe*.
- Proveriti slučaj korišćenja *Pravljenje raspodele (podeli na x, podeli da ima x, manuelno)*.
- Proveriti slučaj korišćenja *Pregled svih studenata jednog smeru*.
- Proveriti slučaj korišćenja *Pretraga studenata jednog smeru*.
- Proveriti slučaj korišćenja *Prikaz globalnog rasporeda studenata jednog smeru*.
- Proveriti slučaj korišćenja *Kopiranje raspodele*.
- Proveriti slučaj korišćenja *Prikaz svih grupa*.
- Proveriti slučaj korišćenja *Izmena raspodele*.
- Proveriti slučaj korišćenja *Izvoz raspodele*.
- Proveriti slučaj korišćenja *Otkazivanje časa*.
- Proveriti slučaj korišćenja *Prikazivanje kombinovanog rasporeda*.
- Proveriti slučaj korišćenja *Dodavanje i brisanje studenata*.
- Proveriti slučaj korišćenja *Dodavanje obaveštenja vezanih za konkretan čas*.
- Proveriti slučaj korišćenja *Dodavanje u grupu*.
- Proveriti slučaj korišćenja *Dodavanje novog predmeta*.
- Proveriti slučaj korišćenja *Unos i ažuriranje podataka o predmetu*.
- Proveriti slučaj korišćenja *Odabir predmeta za koje je zadužen*.
- Proveriti slučaj korišćenja *Unos i izmena kalendara aktivnosti*.

4.3 Testiranje poslovnog ciklusa

- Proveriti posledice operacije dodavanja novog studenta.
- Proveriti posledice operacije dodavanja novog asistenta.
- Proveriti posledice operacije dodavanja novog člana administracije.

- Proveriti posledice kreiranja nove raspodele.

4.4 Testiranje korisničkog interfejsa

- Proveriti jednostavnost navigacije kroz komponente aplikacije.
- Proveriti da li izgled aplikacije odgovara definisanim standardima za izradu korisničkog interfejsa, odnosno da li se slažu sa Material Design specifikacijom koju je razvio Google.
- Proveriti da li korisnički interfejs omogućava jednostavno i intuitivno korišćenje bez potrebe za organizovanjem dodatne obuke.
- Sistem treba da bude lak za korišćenje.
- Sistem treba da bude prilagođen korisnicima koji poseduju dobar nivo znanja rada za računarom.
- Proveriti da li onlajn uputstvo obezbeđuje adekvatnu podršku pri korišćenju naprednijih funkcionalnosti sistema.

4.5 Testiranje performansi

- Proveriti vreme potrebno za prijavljivanje na sistem.
- Proveriti vreme potrebno za kreiranje nove raspodele.
- Proveriti vreme potrebno za generisanje kombinovanog rasporeda.
- Proveriti vreme potrebno za pretragu oglasa.
- Proveriti da li vreme potrebno za pristupanje bazi podataka u cilju izvršenje upita ne prelazi 5 sekundi. Za detalje pogledati dokument [5].

4.6 Testiranje opterećenja

- Proveriti odziv sistema kada mu istovremeno pristupa 500 posetilaca.
- Proveriti odziv sistema kada 120 posetilaca istovremeno pristupa ličnom rasporedu.
- Proveriti odziv sistema kada 85 posetilaca istovremeno pristupa globalnom rasporedu.
- Proveriti odziv sistema kada 120 posetilaca istovremeno pristupa rasporedu aktivnosti.

- Proveriti odziv sistema kada 30 posetilaca istovremeno izvozi raspodelu.
- Proveriti odziv sistema kada 40 posetilaca istovremeno pristupa asistent-skom panelu.
- Proveriti odziv sistema kada 40 posetilaca istovremeno aktivno vrši navigaciju kroz asistentski panel, krećući se kroz smerove, raspodele, grupe i studenete, i izvršavajući akcije poput kreiranje nove raspodele, izmene podataka o grupama i pregled detaljnih podataka o studentima.

4.7 Testiranje kritičnih slučajeva

- Proveriti vreme odziva sistema prilikom prvog korišćenja.
- Proveriti vreme odziva sistema kada 700 posetilaca istovremeno pristupa svom ličnom rasporedu.

4.8 Testiranje sigurnosti i kontrole pristupa

- Proveriti prava pristupa korisnika koji pripadaju različitim grupama (Student, Asistent, Član administracije).

4.9 Testiranje otkaza i oporavka

- Proveriti da li je aplikacija **Raspored** dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 3%. Ovo je u skladu sa ranije definisanim zahtevima u dokumentima [4] i [6].
- Uveriti se da srednje vreme između dva sukcesivna otkaza neće premašiti 120 sati. Ovo je u skladu sa ranije definisanim zahtevom u dokumentu [6].

4.10 Testiranje konfiguracije

- Serverska komponenta sistema treba da funkcioniše na računaru pod operativnim sistemom Linuks.
- Serverska komponenta sistema treba da funkcioniše na računaru pod operativnim sistemom Majkrosoft Vindouz.
- Proveriti da li klijentski deo aplikacije **Raspored** može da se koristi pomoću sledećih veb-čitača: poslednjih nekoliko verzija pregledača *Google Chrome* i

Mozilla FireFox, kao i njihove verzije za mobilne uređaje. Videti dokument [5].

4.11 Testiranje instalacije

Proveriti instalaciju sistema na serveru.

Pošto se radi o veb-aplikaciji, klijenti ne treba da instaliraju nikakav poseban softver, već samo da koriste konekciju ka internetu i već instaliran veb-pregledač (neki od ranije specificiranih).

Glava 5

Strategije testiranja

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

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

5.1 Tipovi testiranja

5.1.1 Testiranje podataka i integriteta baze podataka

<i>Cilj testa</i>	Utvrđiti da metode i procesi za pristup bazi podataka funkcionišu korektno, kao i da nema grešaka u podacima.
<i>Tehnike</i>	Isprobati svaki metod pristupa bazi podataka, za korektne i nekorektne podatke. Isprobati svaki metod za pribavljanje podataka iz baze podataka, za korektne i nekorektne podatke. Ispitati bazu podataka da bi se utvrdilo da se podaci unose kao što se očekuje, da se svi događaji u bazi obrađuju korektno i da se podaci pribavljaju korektno i bez grešaka.
<i>Kriterijum završetka</i>	Metodi i procesi za pristup bazi podataka rade kao što je predviđeno i bez grešaka u podacima.
<i>Napomene</i>	Testiranje zahteva treba obaviti korišćenjem okruženja za rad sa konkretnim DMBS-om da bi se neposredno

uneli i promenili podaci u bazi podataka. Procesi se moraju ručno pokrenuti. Bazu koristiti sa najmanjim mogućim brojem podataka, kako bi se povećala preglednost i lakše uočili neprihvatljivi slučajevi.

5.1.2 Testiranje sistema

<i>Cilj testiranja</i>	Potvrditi korektno kretanje kroz aplikaciju, unos podataka, njihovu obradu i pribavljanje.
<i>Tehnike</i>	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. Utvrditi da li se predviđene poruke o grešci ili upozorenja javljaju kada se radi sa netačnim podacima. Potvrditi da li su poslovna pravila pravilno primenjena.
<i>Kriterijum završetka</i>	Svi planirani testovi su izvršeni. Sve identifikovane greške su adekvatno obrađene.
<i>Napomene</i>	Potrebno je obezbediti pristup serveru na kome će biti instaliran sistem.

5.1.3 Testiranje poslovnog ciklusa

<i>Cilj testiranja</i>	Potvrditi odgovarajuću primenu i procese koji se obavljaju kao podrška zahtevanom modelu poslovanja i utvrđenom rasporedu.
<i>Tehnike</i>	<p>Testiranje simulira poslovne cikluse koji obavljaju sledeće:</p> <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 vremenskom periodu.• Sve funkcije koje su vremenski zavisne treba proveriti korišćenjem tačnih i netačnih 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 javljaju kada se radi sa netačnim podacima. Potvrditi da su poslovna pravila pravilno primenjena.
<i>Kriterijum završetka</i>	Svi planirani testovi su izvršeni. Sve identifikovane greške su adekvatno obrađene.
<i>Napomene</i>	Određivanje datuma i perioda može zahtevati dodatne aktivnosti. Poslovni model bi trebalo da identifikuje odgovarajuće zahteve i procedure koje treba testirati.

5.1.4 Testiranje korisničkog interfejsa

<i>Cilj testiranja</i>	Potvrditi da navigacija kroz aplikaciju korektno odslikava poslovne funkcije i zahteve, uključujući kretanje među komponentama i načine pristupa (ostvariti korisnikova očekivanja o prećicama preko tastature i slično). Potvrditi da su sve komponente korisničkog interfejsa u skladu sa specifikacijom Material Design razvijenu od strane Gugla.
<i>Tehnike</i>	Kreirati ili modifikovati postojeće testove za svaku komponentu da bi se potvrdio očekivan način korišćenja aplikacije.
<i>Kriterijum završetka</i>	Testirane su sve komponente korisničkog interfejsa, jednostavne i intuitivne su za korišćenje, i u potpunosti se slažu sa specifikacijom Material Design.

5.1.5 Testiranje performansi

<i>Cilj testiranja</i>	Potvrditi vreme odziva sistema za napravljene transakcije ili poslovne funkcije za procenjeno prosečno opterećenje i procenjeno maksimalno opterećenje.
<i>Tehnike</i>	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 8jedan korisnik, jedna transakcija) ili ponoviti sa više klijenata (virtuelnih ili stvarnih).
<i>Kriterijum završetka</i>	Jedna transakcija i jedan korisnik: Uspešno izvršavanje test-skripti bez ikakvih padova i za očekivano vreme. Više transakcija i 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 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 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.

5.1.6 Testiranje opterećenja

Cilj testiranja

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

Tehnike

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

Kriterijum završetka

Više transakcija i više korisnika: Uspešno izvršenje test-skripti bez ikakvih padova i za očekivano vreme.

Napomene

Testiranje performansi treba da se obavi na za to spremljenom 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.

5.1.7 Testiranje kritičnih slučajeva

Cilj testiranja

Proveriti da li sistem funkcioniše korektno i bez grešaka u sledećim kritičnim slučajevima.

- malo ili bez memorije na serveru;
- maksimalan broj klijenata (stvarnih ili simuliranih) koji rade na sistemu;
- više korisnika koji zahtevaju istu transakciju nad istim podacima i
- 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 radi ispravno.

Tehnike

Koristiti testove koji su napravljeni za testiranje performansi. Da bi se obavilo testiranje pod ograničenjima, testovi treba da se obavljaju 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 previđene 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 u 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

Cilj testiranja

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.

<i>Tehnike</i>	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.
<i>Kriterijum završetka</i>	Svi planirani testovi se izvršavaju i predviđene granice sistema su dostignute bez pada sistema.
<i>Napomene</i>	Odabrati period vremena koji će biti prihvatljiv za testiranje uslova velikog volumena.

5.1.9 Testiranje sigurnosti i kontrole pristupa

<i>Cilj testiranja</i>	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.
<i>Tehnike</i>	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.
<i>Kriterijum završetka</i>	Za svaki tip korisnika i odgovarajuće funkcije/podatke dozvoljene za korišćenje, sve transakcije funkcionišu na očekivani način, definisane u testovima sistema.

5.1.10 Testiranje otkaza i oporavka

<i>Cilj testiranja</i>	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: <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 i
- 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 sprovedu (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, vrati 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.

<i>Napomene</i>	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

<i>Cilj testiranja</i>	Proveriti i potvrditi da klijentska aplikacija funkcioniše na konfiguracijama na kojima je to predviđeno.
<i>Tehnike</i>	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.
<i>Kriterijum završetka</i>	Za svaku kombinaciju, transakcije iz testa treba da se završe uspešno, bez grešaka.
<i>Napomene</i>	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.

5.2 Alati

U tabli 5.1 su pobrojani alati koji će biti korišćeni za testiranje sistema.

Faza	Alat	Proizvodač	Verzija
Test Management	Rational Unified Process	Rational	TBD
Test Design	Rational Rose	Rational	TBD
Defect Tracking			TBD
Functional Testing			TBD
Performance Testing			TBD
Profiler			TBD
Other			TBD
Project Management	Project, L ^A T _E X	Open Source	TBD
DBMS Tools	SQL Server	SQL Server	TBD

Tabela 5.1: Alati koji će biti korišćeni za testiranje sistema.

Glava 6

Resursi

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

6.1 Radnici

Tabela 6.1 pokazuje predlog rasporeda radnika na aktivnostima testiranja.

6.2 Sistem

U tabeli 6.2 navedeni su sistemski resursi koji će se koristiti za testiranje aplikacije **Raspored**.

6.3 Kontrolne tačke testiranja

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

Tabela 6.3 prikazuje kontrolne tačke testiranja.

6.4 Izlazni produkti testiranja

Rezultati testiranja koji su definisani ovim planom testiranja navedeni su u tabeli 6.4

Radnik i MBR	Komentari
Rukovodilac testiranja: Lazar Ljubenović	Obezbeđuje nadgledanje upravljanjem procesa. Odgovornosti: obezbeđenje tehničke direkcije; prikupljanje odgovarajućih resursa i upravljanje izveštavanjem.
Projektant testova: Lazar Ljubenović	Identifikuje, određuje prioritete i implementira slučajeve testiranja. Odgovornosti: napraviti plan testiranja; napraviti model testiranja i proceniti efektivnost napora uloženog u testiranje.
Tester sistema: Marija Đorđević, Lazar Ljubenović	Izvršiti testiranje. Odgovornosti: izvršiti testiranje, zabeležiti rezultate, popraviti greške, opisati greške.
Administrator testiranja sistema: Marija Đorđević	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: Marija Đorđević	Pripremiti podatke i bazu podataka za testiranje i utvrditi da njihovo stanje odgovara pretpostavkama. Odgovornosti: administrirati podatke, odnosno bazu podataka, za testiranje.
Projektant: Lazar Ljubenović	Identifikovati i definisati operacije, atribute i veze između različitih klasa testova. Odgovornosti: identifikovati i definisati klase testove i identifikovati i definisati pakete testova.
Implementacija: Lazar Ljubenović, Marija Đorđević	Implementirati testove i rasporediti ih u odgovarajuće klase i pakete testova. Odgovornosti: kreirati klase i pakete testova implementirane u test-modelu.

Tabela 6.1: Predlog rasporeda radnika na aktivnostima testiranja. MBR je skraćenica od “minimalni broj radnika”, odnosno predstavlja broj radnika koji će raditi puno radno vreme.

Resurs	Ime / Tip / Serijski broj
Altos SUPRIMO Azdaja	Serijski broj: 474903

Tabela 6.2: Sistemi resurski koji će se koristiti za testiranje.

Zadatak	Trud	Početak	Kraj
Itreacija C1: R1.0 Release <ul style="list-style-type: none"> • Planiranje testova • Projektovanje testova • Razvoj testova • Izvršenje testova • Evaluacija testova 	5	Maj 06 2016	TBD

Tabela 6.3: Kontrolne tačke testiranja.

Dokument	Vlasnik	Ocena/Raspodela	Rok
Plan testiranja	Lazar	Rukovodioci testiranja	TBD
Test okruženje	Marija	-	TBD
Model testiranja	Lazar	Interna ocena	TBD
Skup test-podataka	Marija	Interna ocena	TBD
Test-procedure	Lazar	Interna ocena	TBD
Test-skripte	Marija	-	TBD
Test-podrška	Lazar	-	TBD
Izveštaj o uočenim greškama	Lazar	Interna ocena	TBD
Rezultati testa	Lazar	Interna ocena	TBD
Izvrštaj o evaluaciji testa	Lazar	Interna ocena	TBD

Tabela 6.4: Rezultati testiranja.

6.4.1 Model testiranja

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

6.5 Projektni zadaci

U nastavku 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.