# Blanketomat

# Plan testiranja

# Verzija 1.0

# Istorija revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 11.05.2024. | 1.0 | Inicijalna verzija | Aleksa, Ognjen, Strahinja, Stefan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Sadržaj

1. Ciljevi [4](#__RefHeading___Toc166369058)

2. Opseg [4](#__RefHeading___Toc166369059)

3. Reference [4](#__RefHeading___Toc166369060)

4. Zahtevi za testiranje [4](#__RefHeading___Toc166369061)

5. Strategije testiranja 6

5.1 Tipovi testiranja 6

5.1.1 Testiranje podataka i integriteta baze podataka 6

5.1.2 Testiranje sistema [7](#__RefHeading___Toc166369065)

5.1.3 Testiranje poslovnog ciklusa 7

5.1.4 Testiranje korisničkog interfejsa 8

5.1.5 Testiranje performansi [9](#__RefHeading___Toc166369068)

5.1.6 Testiranje opterećenja 9

5.1.7 Testiranje kritičnih slučajeva 10

5.1.8 Testiranje otkaza i oporavka 10

5.1.9 Testiranje konfiguracije 11

5.2 Alati 12

6. Resursi 12

6.1 Radnici 12

7. Glavne postavke testiranja 14

8. Rezultati testiranja 14

8.1 Model testiranja 14

# PeNcIL

# Ciljevi

Dokument predstavlja sveobuhvatan opis plana testiranja za sistem SSOA. Koncipiran je tako da precizno razloži pristupe i metode verifikacije kako funkcionalnih, tako i nefunkcionalnih zahteva definisanih tokom faza razvoja aplikacije. Primarni ciljevi dokumenta obuhvataju:

* Identifikovati ključne komponente sistema za testiranje na osnovu dostupnih informacija o projektu.
* Izdvojiti ključne testne zahteve iz predloženih konceptualnih smernica.
* Razviti i detaljno opisati strategije testiranja koje će se primenjivati.
* Ustanoviti potrebne resurse za testiranje, te proceniti potrebno vreme i napor za sprovođenje testova.
* Kreirati dokumentaciju koja će sadržati rezultate testiranja.

# Opseg

Plan testiranja fokusira se na integraciju sistema i evaluaciju finalne verzije web aplikacije Blanketomat.

Plan ne uključuje testiranje pojedinačnih komponenti sistema; pretpostavlja se da su programeri već izvršili detaljno testiranje svog koda koristeći metodu crne kutije pre integracije u sistem.

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

# Reference

Reference na osnovu kojih će biti napisan Plan testiranja su:

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

# Zahtevi za testiranje

Ovaj odeljak iznosi elemente sistema koji zahtevaju testiranje, uključujući slučajeve korišćenja, funkcionalne i nefunkcionalne zahteve. Biće opisano samo ono što je potrebno testirati, dok će se detalji o samim testovima naći u kasnijem delu dokumenta pod nazivom Test specifikacija.

**Testiranje podataka i integriteta baze podataka**

Provera otpornosti na SQL injection napade

Provera integriteta podataka prilikom migracije ili ažuriranja

Proveriti mogućnost pristupa bazi Blanketomat.

Proveriti mogućnost istovremenog pristupa istim podacima radi čitanja.

**Testiranje sistema – funkcionalno testiranje**

Proveriti slučaj korišćenja Pregled liste zadataka.

Proveriti slučaj korišćenja Pregled liste pitanja.

Proveriti slučaj korišćenja Pregled spiska studenata.

Proveriti slučaj korišćenja Pregled spiska asistenata

Proveriti slučaj korišćenja Pregled spiska profesora

Proveriti slučaj korišćenja Pregled podataka o određenom studentu.

Proveriti slučaj korišćenja Pregled zadataka po oblastima

Proveriti slučaj korišćenja Pregled zadataka po podoblastima

Proveriti slučaj korišćenja Pregled spiska blanketa

Proveriti slučaj korišćenja Pregled podataka o određenom ispitu

Proveriti slučaj korišćenja Prijavljivanje.

Proveriti slučaj korišćenja Ažuriranje podataka o članu fakulteta.

Proveriti slučaj korišćenja Dodavanje novog zadatka.

Proveriti slučaj korišćenja Brisanje postojećeg pitanja.

Proveriti slučaj korišćenja Ažuriranje osnovnih podataka o predmetu.

Proveriti slučaj korišćenja Kreiranje novog studenta.

Proveriti slučaj korišćenja Brisanje postojećeg člana fakulteta.

Proveriti slučaj korišćenja Arhiviranje postojećeg člana fakulteta.

**Testiranje poslovnog ciklusa**

Proveriti posledice operacije ažuriranja postojećeg člana fakulteta.

Proveriti posledice operacije dodavanja novog člana fakulteta.

Proveriti posledice operacije brisanja zadatka.

**Testiranje korisničkog interfejsa**

Provera intuitivnosti i jednostavnosti korišćenja

Provera konzistentnosti korisničkog interfejsa

Proveriti da li korisnički interfejs prilagodjen članovima fakulteta za jednostavno korišćenje i manipulisanje sadržajem bez potrebe za organizovanjem dodatne obuke. (Specifikacija zahteva, odeljak 7.2).

**Testiranje performansi**

Proveriti vreme potrebno za prijavljivanje na sajt.

Proveriti vreme potrebno za dodavanje novog člana fakulteta.

Proveriti vreme potrebno za dodavanje blanketa.

Proveriti da li vreme potrebno za pristupanje bazi podataka u cilju izvršenje upita ne prelazi 3 sekundi (Specifikacija zahteva, odeljak 7.4).

**Testiranje opterećenja**

Proveriti odziv sistema kada mu istovremeno pristupa 200 posetilaca.

Proveriti odziv sistema kada 100 posetilaca istovremeno pristupa stranici Pregled blanketa.

Proveriti da li sistem može da podrži do 500 simultanih pristupa korisnika portalu (Arhitekturni projekat, odeljak 11).

**Testiranje kritičnih slučajeva**

Proveriti vreme odziv sajta prilikom prvog korišćenja.

Proveriti vreme odziva sistema kada 100 posetilaca istovremeno pristupa stranici Pregled banketa.

**Testiranje otkaza i oporavka**

Proveriti da li je Blanketomat sajt dostupan 24 časa dnevno, 7 dana u nedelji. (Arhitekturni projekat, odeljak 12)

Proveriti da li srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 168sati (Arhitekturni projekat, odeljak 12).

**Testiranje konfiguracije**

Klijentska komponenta sistema treba da funkcioniše na računaru koji podržava moderne web browser i pristup internetu.(Specifikacija zahteva, odeljak 7.6).

# Strategije testiranja

Strategije testiranja pružaju smernice o metodama koje se primenjuju pri testiranju softverskih aplikacija. Dok prethodni odeljak o zahtevima za testiranje specificira elemente koji će biti podvrgnuti testiranju, ovaj odeljak će detaljno objasniti pristupe kojima će se ti elementi testirati. Temelj svake strategije testiranja su tehnike koje se upotrebljavaju i kriterijumi koji određuju kada je proces testiranja uspešno završen.

## Tipovi testiranja

### Testiranje podataka i integriteta baze podataka

|  |  |
| --- | --- |
| Cilj Testa: | • Potvrditi da baza podataka čuva podatke tačno i dosledno kroz različite scenarije upotrebe i da sistem može adekvatno rukovati greškama u podacima bez gubitka integriteta. Osigurati da svi podaci uneti u bazu podataka ostaju tačni i neizmenjeni tokom različitih transakcija. Proveriti mehanizme za očuvanje integriteta podataka kao što su ograničenja stranih ključeva, ograničenja jedinstvenosti, i provere pri unosu podataka. |
| Tehnike: | * Koristiti skripte i alate za automatsko testiranje koji mogu generisati i manipulisati podacima na načine koji testiraju ograničenja i integritet baze. * Simulirati scenarije koji uključuju neispravne ili konfliktne podatke kako bi se proverila robustnost mehanizama za održavanje integriteta. * Primeniti reviziju podataka kako bi se proverila tačnost i korektnost podataka unesenih u bazu. |
| Kriterijum završetka: | • Svi testovi su izvršeni i sistem je pokazao sposobnost da održava integritet podataka čak i u slučajevima kada su uneti neispravni ili konfliktni podaci.. |
| Napomene: | • Testirati različite tipove baza podataka (relacione, NoSQL, itd.) kako bi se osigurala širina testiranja integriteta. |

### Testiranje sistema

|  |  |
| --- | --- |
| Cilj testiranja: | • Verifikovati ispravnost navigacije kroz aplikaciju, tačnost unosa i obrade podataka, te pravilnost izvlačenja informacija. |
| Tehnike: | * Simulirati korisničke interakcije kako bi se testiralo sve definisane tokove u aplikaciji koristeći kako validne, tako i nevalidne podatke. * Proveriti ispravnost rezultata dobijenih obradom tačnih podataka. * Kontrolisati pojavljivanje grešaka ili upozorenja prilikom unosa neispravnih podataka. * Analizirati primenu poslovnih pravila u različitim scenarijima upotrebe. |
| Kriterijum završetka: | * Uspešno izvedeni svi predviđeni test scenariji. * Adekvatno rešene sve uočene greške i nepravilnosti. |
| Napomene: | * Obezbediti konstantan pristup razvojnom serveru tokom testiranja kako bi se omogućilo neometano izvođenje testova. |

### Testiranje poslovnog ciklusa

|  |  |
| --- | --- |
| Cilj testiranja: | • Osigurati da web aplikacija efikasno podržava ključne poslovne procese prema definisanim operativnim standardima i vremenskim okvirima. |
| Tehnike: | * Implementacija scenarija koji emuliraju realne poslovne operacije kako bi se testirala aplikacija pod opterećenjem sličnim stvarnom radnom okruženju. * Modifikacija postojećih testova za funkcionalnost kako bi se uključili zahtevi za višestrukim simultanim pristupom korisnika, simulirajući tako realističniju upotrebu. * Verifikacija vremenski osetljivih funkcija kako bi se potvrdilo da se aktivnosti pokreću u predviđenim intervalima. * Analiza ishoda obrade korektnih podataka te identifikacija i rešavanje grešaka u obradi nekorektnih podataka. * Provera usklađenosti operacija sa poslovnim pravilima i regulativama. |
| Kriterijum završetka: | * Uspešno izvedeni svi planirani testovi. * Identifikovane i korigovane sve anomalije i problemi u funkcionisanju. |
| Napomene: | * Planiranje testiranja može zahtevati prilagođavanje kalendara i vremenskih okvira za određene poslovne aktivnosti. * Nužno je definisanje preciznih parametara poslovnog modela i procesa koji su predmet testiranja. |

### Testiranje korisničkog interfejsa

|  |  |
| --- | --- |
| Cilj testiranja: | * Proveriti da li korisnički interfejs aplikacije omogućava jasnu, intuitivnu interakciju sa korisnicima i da su svi elementi interfejsa funkcionalni i estetski prihvatljivi. |
| Tehnike: | * Primena testova koji simuliraju korisničke interakcije kako bi se proverila odzivnost, funkcionalnost i estetika svih komponenti korisničkog interfejsa. * Korišćenje alata za automatizovano testiranje interfejsa koji detektuje probleme u navigaciji, prikazu elemenata i interakciji sa korisnicima. * Testiranje interfejsa na različitim uređajima i rezolucijama ekrana da bi se osigurala konzistentnost i prilagodljivost na različitim platformama. * Evaluacija pristupačnosti interfejsa za korisnike sa posebnim potrebama, uključujući testiranje čitljivosti, dostupnosti i navigacije. |
| Kriterijum završetka: | * Interfejs uspešno zadovoljava sve zahtevane kriterijume funkcionalnosti, odzivnosti, estetike i pristupačnosti. * Identifikovani svi problemi su efikasno rešeni i verifikovani kroz ponovljene testove. |
| Napomene: | • Potrebno je kontinuirano ažuriranje testova kako bi se pratili najnoviji standardi dizajna i tehnologije interfejsa. |

### Testiranje performansi

|  |  |
| --- | --- |
| Cilj testiranja: | • Ocena efikasnosti sistema u odnosu na vreme odziva pri različitim nivoima opterećenja, uključujući procenjeno prosečno i maksimalno opterećenje. |
| Tehnike: | * Izvoditi testove izvedene iz testiranja poslovnog modela, prilagođene za evaluaciju performansi pri varijabilnim opterećenjima. * Modifikacija skripti i podataka da bi se povećala frekvencija transakcija i simuliralo realno povećanje opterećenja na sistemu. * Testovi se izvode na pojedinačnom sistemu za pojedinačne transakcije, kao i na višekorisničkim platformama kako bi se simulirala stvarna upotreba. |
| Kriterijum završetka: | * Pojedinačne transakcije: Efikasno izvršenje bez pada sistema i unutar očekivanih vremenskih okvira. * Multikorisničke transakcije: Stabilnost sistema pri visokom opterećenju sa zadržavanjem odgovarajućeg vremena odziva. |
| Napomena: | * Pri testiranju performansi koristiti različite tehnike za generisanje opterećenja, uključujući direktno izvršavanje transakcija na serveru, kreiranje virtualnih korisnika, i korišćenje realnih korisnika sa različitih lokacija. * Za potrebe testiranja koristiti bazu podataka optimizovanu za očekivanu stvarnu upotrebu kako bi se osigurala realistična simulacija radnog okruženja. |

### Testiranje opterećenja

|  |  |
| --- | --- |
| Cilj testiranja: | • Evaluacija sposobnosti sistema da održava optimalno vreme odziva i stabilnost pod različitim nivoima opterećenja, od prosečnog do maksimalnog. |
| Tehnike: | * Izvođenje testova dizajniranih za procenu opterećenja, koristeći testne scenarije iz poslovnog ciklusa. * Povećanje broja transakcija i intenziviranje izvršenja skripti za simulaciju različitih nivoa opterećenja u stvarnom radnom okruženju. |
| Kriterijum završetka: | * Adekvatno izvršenje testova bez sistema pada i održavanje vremena odziva unutar zadatih parametara, čak i pod maksimalnim opterećenjem. |
| Napomene: | * Testiranje treba obaviti na specijalizovanim testnim mašinama kako bi se osigurala preciznost i kontrola rezultata. * Veličina i konfiguracija testne baze podataka moraju odražavati realne uslove upotrebe kako bi se rezultati testa smatrali validnim. |

### Testiranje kritičnih slučajeva

|  |  |
| --- | --- |
| Cilj testa: | • Provera stabilnosti i funkcionalnosti sistema pod ekstremnim uslovima, kao što su ograničeni resursi memorije, maksimalan broj korisnika, i intenzivno korišćenje podataka. |
| Tehnike: | * Korišćenje modifikovanih performans testova za simuliranje kritičnih situacija. * Realizacija testova na konfiguracijama sa smanjenim kapacitetima memorije i povećanim brojem transakcija kako bi se procenila granica izdržljivosti sistema. |
| Kriterijum završetka: | • Sistem treba da funkcioniše bez prekida unutar predviđenih teških uslova, ili se precizno identifikuju okolnosti koje dovode do sistema otkaza. |
| Napomene: | * Potrebno je koristiti specijalizovane alate za generisanje visokog opterećenja i kontrolisanu redukciju dostupne memorije. |

### Testiranje otkaza i oporavka

|  |  |
| --- | --- |
| Cilj testiranja: | • Proveriti sposobnost sistema da se oporavi i nastavi sa radom nakon niza kritičnih kvarova, uključujući prekide napajanja, mrežne prekide i greške na hardveru, osiguravajući da se sistem može pouzdano vratiti u normalno operativno stanje. |
| Tehnike: | * Primena testova iz poslovnog i funkcionalnog testiranja za postavljanje scenarija prije izazivanja kvarova. * Simulacija prekida napajanja na serverima i klijentskim uređajima za testiranje automatskog oporavka. * Ograničavanje mrežne povezanosti za procenu sposobnosti sistema da se nosi sa gubicima u komunikaciji. * Izazivanje hardverskih kvarova na skladišnim uređajima kako bi se testirala robusnost mehanizama za oporavak podataka. * Manipulacija podacima u bazi da bi se testirala pravilnost rekonstrukcije podataka nakon kvarova. |
| Kriterijum završetka: | • Sistem treba efikasno obnoviti sve funkcije i podatke nakon svakog testiranog scenarija kvara, bez trajnih oštećenja ili gubitka podataka. |
| Napomena: | * Treba provesti niz rigoroznih testova u kontrolisanom okruženju da bi se osiguralo da su svi aspekti sistema otporni na očekivane i neočekivane prekide. |

### Testiranje konfiguracije

|  |  |
| --- | --- |
| Cilj testiranja: | • Osigurati da klijentska aplikacija ispravno radi na svim zvanično podržanim konfiguracijama uređaja i operativnih sistema. |
| Tehnike: | * Primena kombinacije funkcionalnih testova i testova integracije za utvrđivanje kompatibilnosti aplikacije sa različitim tehničkim okruženjima. * Simuliranje tipičnog korisničkog okruženja aktiviranjem i deaktiviranjem raznih aplikacija pre i tokom testiranja. * Provođenje tipičnih transakcija unutar aplikacije da bi se ocenilo kako se ponaša u realnim uslovima. * Testiranje aplikacije pri smanjenim resursima RAM-a da bi se analizirala njena efikasnost kada su resursi ograničeni. |
| Kriterijum završetka: | • Aplikacija mora bez grešaka da funkcioniše na svim predviđenim konfiguracijama, potvrđujući njenu stabilnost i pouzdanost. |
| Napomene: | * Specifikacija aplikacija koje se redovno koriste na podržanim uređajima. * Definisanje količine i vrste podataka s kojima aplikacija operiše. * Potrebno je detaljno opisati mrežnu i server infrastrukturu koja podržava klijentsku aplikaciju. |

## 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 |
| Functional Testing |  |  | TBD |
| Performance Testing |  |  | TBD |
| Profiler |  |  | TBD |
| Project Management | Project  Word  Excel | Microsoft  Microsoft  Microsoft | TBD |
| DBMS tools | Microsoft SQL Server | Microsoft SQL Server | 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 | Strahinja, Aleksa | Obezbeđuje nadgledanje upravljanjem procesa  Odgovornosti:   * Obezbediti tehničku direkciju * Prikupljanje odgovarajućih resursa * Upravljanje izveštavanjem |
| Projektant testova | Strahinja, Aleksa | Identifikuje, određuje prioritete i implementira slučajeve testiranja  Odgovornosti:   * Napraviti plan testiranja * Napraviti model testiranja * Proceniti efektivnost napora uloženog u testiranje. |
| Tester sistema | Ognjen, Stefan | Izvršiti testiranje  Odgovornosti:   * Izvršiti testiranje * Zabeležiti rezultate * Popraviti greške * Opisati greške |
| Administrator testiranja sistema | Strahinja | 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 | Aleksa | Pripremiti podatke i bazu podataka za testiranje i utvrditi da njihovo stanje odgovara pretpostavkama.  Odgovornosti:   * Administrirati podatke (bazu podataka) za testiranje |
| Projektant | Strahinja | Identifikovati i definisati operacije, atribute i veze između različitih klasa testova  Odgovornosti:   * Identifikovati i definisati klase testova * Identifikovati i definisati pakete testova |
| Implementacija | Ognjen, Stefan | Implementirati testove i rasporediti ih u odgovarajuće klase i pakete testova.  Odgovornosti:   * Kreirati klase i pakete testova implementirane u test modelu. |

# Kontrolne tačke testiranja

Aktivnosti testiranja su usko povezane sa iteracijama u procesu razvoja softvera. Razvojni ciklus web sajta Blanketomat biće realizovan kroz jednu iteraciju, koja će obuhvatiti celokupni ciklus testiranja — od planiranja i projektovanja, preko razvoja i izvršavanja, do evaluacije.

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 | 10 | Maj 05 | Maj 12 |

# Izlazni produkti testiranja

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Dokument** | **Vlasnik** | **Ocena / Raspodela** | **Due Date** |
| Plan Testiranja | Strahinja | Rukovodioci testiranja | Maj 06 |
| Test okruženje | Strahinja | - | Maj 06 |
| Model testiranja | Strahinja, Aleksa | Interna ocena | Maj 06 |
| Skup test podataka | Aleksa | Interna ocena | Maj 09 |
| Test procedure | Ognjen | Interna ocena | Maj 09 |
| Test skripte | Stefan | - | Maj 09 |
| Test podrška | Strahinja | - | Maj 09 |
| Izveštaj o uočenim greškama | Ognjen, Stefan | Rukovodioci testiranja | Maj 12 |
| Rezultati testa | Ognjen | Rukovodilac testiranja | Maj 12 |
| Izveštaj o evaluaciji testa | Strahinja | Rukovodilac testiranja | Maj 12 |

## Model testiranja

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