

*Prolećni semestar, 2022/23*

*PREDMET: SE325 - Upravljanje projektima razvoja softvera*

Test plan

Profesor: **Nemanja Zdravković**

Student: **Dušan Stanković**

Asistent: **Sara Nikolić**

Indeks: **3611**

Niš, 2023. god

Sadržaj

[1. Uvod 3](#_Toc142694489)

[2. Ciljevi i zadaci 3](#_Toc142694490)

[2.1 Ciljevi 3](#_Toc142694491)

[2.2 Zadaci 4](#_Toc142694492)

[3. Strategija testiranja 4](#_Toc142694493)

[3.1 Junit testiranje 4](#_Toc142694494)

[3.2 Mockup testiranje 4](#_Toc142694495)

[3.3 Integraciono testiranje 4](#_Toc142694496)

[3.4 Beta testiranje 5](#_Toc142694497)

[4. Hardverkse potrebe 6](#_Toc142694498)

[5. Potrebe okoline 6](#_Toc142694500)

[6. Vreme testiranja 6](#_Toc142694501)

[Za testiranje je planirano ukupno mesec ipo dana. 6](#_Toc142694502)

[7. Funkcionalnosti koje će biti testirane 6](#_Toc142694503)

[8. Funkcionalnosti koje neće biti testirane 6](#_Toc142694504)

[9. Alati 6](#_Toc142694505)

[10. Odobrenja 7](#_Toc142694506)

# Uvod

Testiranje je jednako ključno kao i sam proces razvoja softvera, jer nam omogućava da potvrdimo da naš sistem ispravno funkcioniše i ispunjava sve potrebe za koje je namenjen. Da bismo uspešno sproveli testiranje, prvo je neophodno napraviti detaljan plan koji će definisati korake i pristup testiranju. Ovaj plan će uključivati postavljanje ciljeva testiranja, identifikaciju ključnih zadataka koji će biti obuhvaćeni testiranjem, kao i jasno određivanje onih koji neće biti obuhvaćeni testovima.

Pored toga, planiranje testiranja će uključivati razmatranje potrebnih hardverskih i softverskih komponenti koje će biti korišćene u procesu testiranja. Izbor odgovarajućih alata za testiranje takođe je važan deo plana, jer će nam pružiti sredstva za efikasno i pouzdano testiranje softvera.

Jednako je važno osigurati da imamo dovoljno resursa, kao što su odgovarajući broj članova tima, vreme i budžet, kako bi testiranje bilo sprovedeno u skladu sa planom i sa potrebnom pažnjom.

U celini, planiranje testiranja je ključni korak kako bismo obezbedili da softver bude kvalitetno i pouzdano testiran pre nego što bude pušten u produkciju. To nam omogućava da identifikujemo i rešimo probleme pre nego što postanu ozbiljni, čime se poboljšava kvalitet i performanse softverskog sistema.

# Ciljevi i zadaci

## Ciljevi

Testiranje je ključni deo procesa razvoja softvera koji zahteva pažnju i angažovanje svih članova tima, uključujući i testere i programere. Cilj testiranja je identifikovati potencijalne greške i nesavršenosti u softverskom rešenju kako bi se osiguralo da sistem funkcioniše ispravno i pouzdano u interakciji sa korisnicima.

Svi članovi tima, uključujući programere, imaju odgovornost da testiraju svoj kod i funkcionalnosti koje su razvili. Programeri su najbliži delu softvera koji su kreirali, stoga su oni prvi koji mogu primetiti eventualne greške i nedostatke. To omogućava brzu reakciju i ispravku problema pre nego što dođe do značajnijih komplikacija.

Testiranje takođe zahteva pažnju korisničkog iskustva i ponašanja korisnika prilikom interakcije sa sistemom. Ovo je ključno za prilagođavanje softvera kako bi se korisničko iskustvo poboljšalo i omogućilo optimalno korišćenje sistema.

Sveukupno, testiranje je iterativni proces koji se kontinuirano odvija tokom razvoja softvera. Angažovanje svih članova tima i konstantno praćenje performansi sistema omogućava identifikaciju i rešavanje problema u ranoj fazi, što doprinosi kvalitetu i uspehu projekta.

## Zadaci

Zadaci koji treba da se obave tokom testiranja su:

* Testiranje funkcionalnosti
* Testiranje korisničkog interfejsa
* Beta testiranje
* Integraciono testiranje

# Strategija testiranja

## Junit testiranje

**Opis:** Svaki član razvojnog tima ima svoju odgovornost kada je u pitanju testiranje. Programeri koji razvijaju sistem su odgovorni za Unit testiranje, što znači da svaka funkcionalnost koju razvijaju mora biti pokrivena odgovarajućim Unit testovima. Ovo osigurava da svaki deo koda bude temeljno testiran kako bi se otkrile moguće greške i nedostaci.

Unit testovi su automatski testovi koji proveravaju određene jedinice koda, tj. manje delove softverskog rešenja, kako bi se obezbedilo da svaka funkcionalnost radi ispravno. Svaka funkcionalnost mora biti pokrivena Unit testovima kako bi se osigurala njena funkcionalnost i integritet.

**Učesnici:** Dušan Stanković (backend developer) i Marko Marković (frontend developer)

## Mockup testiranje

**Opis:** Za proveru komunikacije između frontend i backend dela, koristićemo mock testiranje. Ova vrsta testiranja je izuzetno korisna jer nam omogućava da simuliramo rad aplikacije bez pristupa stvarnoj bazi podataka klijenata. Na taj način možemo proveriti kako frontend reaguje na podatke koje dobija od backenda, bez rizika da utičemo na stvarne podatke ili da narušimo integritet baze.

Mock testiranje se koristi za izolovanje različitih delova aplikacije kako bi se proverila njihova funkcionalnost nezavisno od drugih delova sistema. Ovo nam omogućava da identifikujemo i ispravimo greške i nedostatke pre nego što aplikacija bude u potpunosti integrisana.

**Učesnici:** Nikola Nikolić (Tester)

## Integraciono testiranje

**Opis:** U cilju sveobuhvatnog testiranja softvera, primenićemo integraciono testiranje koristeći bottom-up tehniku. Ova metoda podrazumeva početak testiranja od najmanjih modula softvera, a zatim postepeno prelazimo na testiranje celokupne aplikacije. Ovaj pristup donosi nekoliko ključnih prednosti koje doprinose kvalitetu i efikasnosti testiranja.

Prvo, bottom-up pristup omogućava brže otkrivanje grešaka. Fokusiranje na manje module omogućava nam da lakše primetimo eventualne nedostatke i probleme u funkcionalnostima, jer su obim i kompleksnost manji u odnosu na celokupnu aplikaciju. Na taj način, greške se mogu identifikovati i rešiti pre nego što se prošire na veći deo softverskog sistema.

Drugo, ovaj pristup olakšava i vremensko i finansijsko ispravljanje grešaka. Početak testiranja od manjih modula omogućava nam da greške otkrijemo i ispravimo u ranim fazama razvoja, pre nego što se aplikacija u potpunosti integriše. Time se smanjuje rizik od kritičnih problema koji bi mogli da zaustave razvoj ili iziskuju velike resurse za ispravku.

Treće, bottom-up tehnikom možemo postepeno integrisati i testirati različite delove aplikacije kako se razvijaju, što omogućava kontinuirano testiranje i proveru kako novododati moduli utiču na postojeći softver. Ovo pruža bolji uvid u celokupan sistem i pomaže u otkrivanju potencijalnih konflikata i problema tokom integracije.

Koristeći ovu metodu integracionog testiranja, možemo obezbediti stabilniju, pouzdaniju i visokokvalitetniju aplikaciju koja zadovoljava potrebe korisnika i očekivanja naručioca.

**Učesnici:** Nikola Nikolić (Tester)

## Beta testiranje

**Opis:** Nakon što smo završili proces testiranja i ispravili sve identifikovane greške u sistemu, omogući ćemo korisniku da testira softver u realnom okruženju. Korisnik će dobiti određene zadatke koje treba da izvrši, dok ćemo mi pratiti njegov napredak i reakcije na softver.

Ako korisnikov rad i rezultati testiranja softvera odgovaraju očekivanjima i funkcionalnostima sistema, tada ćemo ga pustiti u produkciju. Međutim, ako se pojave bilo kakva neslaganja ili neusklađenosti sa očekivanjima, pažljivo ćemo dokumentovati te slučajeve.

Nakon dokumentacije problema, tim će se fokusirati na ispravljanje i korekciju identifikovanih neusaglašenosti. Potrebne izmene će biti implementirane u softveru, a zatim će se ponovo sprovesti testiranje kako bismo potvrdili da su sve korekcije uspešno rešile prethodne probleme.

Samo nakon što ponovno testiranje pokaže zadovoljavajuće rezultate i potvrđeno je da su svi problemi ispravljeni, možemo s punim poverenjem pustiti softver u rad i omogućiti korisnicima da ga koriste u stvarnom okruženju. Ovaj pristup nam omogućava da obezbedimo visok nivo pouzdanosti, kvaliteta i funkcionalnosti softverskog sistema pre nego što postane dostupan širem krugu korisnika.

**Učesnici:** Krajnji korisnici sistema

# Hardverkse potrebe

# Softver je dizajniran i razvijan da podržava testiranje na Windows operativnim sistemima u rasponu od Windows 8.1 do najnovije verzije Windows 11. Takođe, podržan je i za testiranje na mobilnim uređajima sa Android operativnim sistemom, počevši od verzije Android 8 pa sve do najnovije verzije Android 14.

# Potrebe okoline

Aplikacija zahteva internet konekciju za korišćenje na mobilnim uređajima i podržava Android verzije od Android 8 do najnovije verzije. Dizajnirana je da se prilagodi različitim dimenzijama Android uređajima kako bi korisnicima pružila optimalno iskustvo.

# Vreme testiranja

# Za testiranje je planirano ukupno mesec ipo dana.

# Funkcionalnosti koje će biti testirane

* Prijavljivanje na sistem
* Kreiranje naloga
* Pregled prisustva
* Pregled predispitnih obaveza
* Unos prisustva
* Unos predispitnih obaveza

# Funkcionalnosti koje neće biti testirane

Kako testiranje oduzima jako puno vremena i resursa, nisu potrebne da se testiraju sve funkcionalnosti. Neće se vršiti pojedinačno testiranje funkcionalnosti koje su međusobno povezane.

# Alati

Mockito- Za testiranje frontend dela sistema.

Hotjar- Za testiranje korisničkg interfejsa kod beta testinga.

Jmockit- Za testiranje funkcionalnosti.

# Odobrenja

Za svo testiranja odgovoran je glavni tester. Sva odobrenja koja se tiču testova vrši lider projekta Dušan Stanković.