



Naziv Projekta	Domaći Zadatak 07 - DZ07
Naziv Proizvoda	Aplikacija za vođenje finansija
Vezija Proizvoda:	1.0

Master Test Plan

Id broj dokumenta: MTP-1.0

Polaznik: Branislav Manojlović
Datum: 18.05.2018.

Sadržaj:

Uvod	3
1. Analiza Proizvoda	3
2. Funkcionalnosti planirane za testiranje	4
3. Strategija testiranja	5
4. Faze testiranja	5
4.1. Faza I – izrada plana testiranja	5
4.2. Faza II – definisanje test slučajeva.....	5
4.3. Faza III – dizajniranje testova.....	5
4.4. Faza IV – sprovođenje Jediničnog testiranja.....	5
4.5. Faza V – sprovođenje Integracionog testiranja.....	5
4.6. Faza VI – sprovođenje Sistemskog testiranja.....	6
4.7. Faza VII – testiranje Sigurnosti.....	6
4.8. Faza VIII – Beta testiranje.....	6
5. Test Podaci	6

Uvod

Predmet testiranja je Aplikacija koja će služiti za vođenje finansija u malim zajednicama (porodicama).

Potrebno je sprovesti testiranje softverskog proizvoda radi provere ispravnosti softvera koji se razvija i radi uočavanja defekata, ukoliko postoje.

Testiranje softverskog proizvoda će se izvršiti po fazama definisanim u Master Test Planu, pri čemu će nakon svake faze testiranja biti izvršeno Regresivno testiranje, koje podrazumeva ponavljanje testova koji su već izvršeni u ranijim iteracijama kako bi se proverilo da li ranije implementirane funkcionalnosti i dalje rade ispravno.

1. Analiza Proizvoda

Naziv softverskog proizvoda: "Aplikacija za vođenje finansija".

Verzija: 1.0

Osnovne korisničke uloge koje postoje u aplikaciji su: korisnik – roditelj i korisnik – dete. Na samom početku korisnik mora da se uloguje kao dete ili kao uređivač aplikacije odnosno roditelj. Za ova dva klijenta biće razvijen zaseban izgled aplikacije. Roditelj će moći da se uloguje sa korisničkim podacima, zatim unese finansije ili obaveze, kao i dodeljuje role korisniku. Potrebno je uneti podatke kao što su suma, opis, datum izvršenja. Sa druge strane, korisnik odnosno dete moći će da pregleda svoje finansije i obaveze, menja obaveze, dodaje sebi obaveze i šalje zahtev za finansije. Roditeljski nalog može da odobri ili odbije zahtev. Roditelj će imati uvid u sve finansije dečijih naloga, kako bi mogao da rasporedi budžet.

Cilj izrade softvera je da omogući manjim zajednicama lakše vođenje finansija, kao i štednju.

Funkcionalni zahtevi za korisnika roditelja su:

- a) Roditelju treba omogućiti da se uloguje sa korisničkim podacima,
- b) Roditelju treba omogućiti da unese finansije (suma),
- c) Roditelju treba omogućiti da unese obaveze (opis i datum izvršenja),
- d) Roditelju treba omogućiti da dodeljuje role korisniku-detetu,
- e) Roditelju treba omogućiti da može da primi zahtev za finansijama upućen od korisnika/deteta,
- f) Roditelju treba omogućiti da može da odbije zahtev upućen od korisnika/deteta,
- g) Roditelju treba omogućiti da može da odobri zahtev upućen od korisnika/deteta,
- h) Roditelju treba omogućiti da ima uvid u sve finansije dečijih naloga.

Funkcionalni zahtevi za korisnika dete su:

- a) Detetu treba omogućiti da može da pregleda svoje finansije,
- b) Detetu treba omogućiti da može da pregleda svoje obaveze,
- c) Detetu treba omogućiti da može da menja svoje obaveze,
- d) Detetu treba omogućiti da može da dodaje svoje obaveze,
- e) Detetu treba omogućiti da može da šalje zahtev za finansije roditelju.

Zahtevi okruženja u kome softver treba da radi:

- a) Specifikacijom nije precizirano za koju platformu će biti razvijan softverski proizvod, ali može se uzeti kao pretpostavka da je softverski proizvod namenjen da se razvija kao web aplikacija.

Zahtevi u pogledu performansi:

- a) Pošto je aplikacija namenjena da se koristi u okviru manjih grupa korisnika, same performanse koje se očekuju od aplikacije u pogledu hendlovanja broja korisnika, količine unetih podataka, rada i komunikacije sa mrežom su skromne. Odnosno, aplikacija je namenjena da radi u sistemima osnovnih LAN (*Local Area Network*) mreža i samim tim i u uslovima manjeg saobraćaja.

Forme koje aplikacija poseduje obuhvataju:

- a) Login forma,
- b) Forma za kreiranje profila,
- c) Forma za unos podataka (finansija, obaveza)
- d) Forma za pregled obaveza,
- e) Forma za pregled raspoloživih finansija.

2. Funkcionalnosti planirane za testiranje

Na osnovu Analize softverskog proizvoda, izdvojene su osnovne funkcionalnosti koje imaju biti testirane, a u njih spadaju:

- Funkcionalnost kreiranja korisničkog profila,
- Funkcionalnost logovanja korisnika na aplikaciju,
- Funkcionalnost unosa podataka od strane korisnika,
- Funkcionalnost izmene unetih podataka od strane korisnika u skladu sa korisničkim ulogama,
- Funkcionalnost obrade unetih podataka u bazu u smislu sumiranja unetih podataka,
- Funkcionalnost čuvanja korisničkih podataka u bazi,
- Funkcionalnost pravilnog prikazivanja raspoloživih podataka korisnicima

3. Strategija testiranja

Testiranje softverskog proizvoda će se izvršiti primenom više strategija testiranja.

Primeniće se:

- funkcionalno jedinično testiranje strategijama Crne kutije (Black box), zato što je potrebno znati šta pojedine komponente softvera rade,
- integraciono testiranje strategijama Bele kutije (White box), pošto je potrebno znati na koji način su komponente povezane prilikom integracije.

4. Faze testiranja

4.1. Faza I – izrada plana testiranja.

Potrebno je detaljno definisati plan testiranja softverskog proizvoda. Predviđeno vreme trajanja Faze I je 5 radnih dana.

4.2. Faza II – definisanje Test slučajeva.

Potrebno je prema osnovnim funkcionalnostima aplikacije definisati Test slučajeve koji se imaju sprovesti u samom postupku testiranja. Predviđeno vreme trajanja Faze II je 5 radnih dana.

4.3. Faza III – dizajniranje testova.

Potrebno je izvršiti dizajniranje testova, kojim će se izvršiti Test slučajevi definisani u Fazi II. Za proveru ispravnosti softverskog proizvoda, potrebno je dizajnirati JUnit testove, kojima će se testirati pojedinačne metode svake klase u aplikaciji, čime će se proveriti ispravnost svake pojedinačne komponente u sistemu, čak i pre same međusobne integracije i povezivanja ovih komponenti. Potrebno je dizajnirati Integracione testove kojima će se proveriti da li se pojedinačne komponente koje čine sistem međusobno ispravno uklapaju i čine jedan funkcionalan sistem. Na kraju potrebno je dizajnirati Sistemske testove, kojima se testira ispravnost celokupnog sistema kao jedne celine s kraja na kraj (end-to-end). Predviđeno vreme trajanja Faze III je 15 radnih dana.

4.4. Faza IV – Sprovođenje Jediničnog testiranja.

Ovo je faza u kojoj se sprovode JUnit testovi. Ovim testovima vrši se provera da li su metode u klasama aplikacije kreirane u skladu sa planom. Proverava se da li metode za validne ulazne podatke daju očekivane izlazne podatke, odnosno rezultate. Tokom ovog testiranja pišu se test metode koje ispituju validne i nevalidne test slučajeve, za svaku od kreiranih funkcionalnih metoda aplikacije.

Na ovaj način će se testirati svaka komponenta u softverskom proizvodu koja se može posmatrati kao nezavisna i samostalna celina i koja se može izvući iz sistema i testirati izolovano od ostalih komponenti. Na ovaj način će se prvo testirati Forme aplikacije. Predviđeno vreme trajanja ove faze je 10 radnih dana.

4.5. Faza V – Sprovođenje Integracionog testiranja.

Ovo je faza u kojoj se sprovodi Integraciono testiranje, kao nastavak Jediničnog testiranja. Kada su napravljene komponente i kada je u Jediničnom testiranju provereno da one rade kao pojedinačne celine potpuno ispravno, ona je potrebno testirati ih u radu povezane sa ostalim komponentama. Na ovaj način se proverava tačnost toka i razmene podataka, potencijane smetnje u komunikaciji, problemi u interfejsima za razmenu podataka i slično. Predviđeno vreme trajanja ove faze je 10 dana.

4.6. Faza VI – Sprovođenje Sistemskog testiranja

Ovo je faza u kojoj se sprovodi Sistemsko testiranje. Ovo je testiranje sistema koji je u potpunosti integrisan. Ovde se vrši testiranje komponenti koje nemaju direktne veze jedna sa drugom i ne komuniciraju direktno jedna sa drugom ali imaju neku zavisnost sa podacima.

Tokom ovog testiranja, sprovedeće se Parcijalno sistemsko testiranje (*Smoke test*) u kome će se prvo proveriti samo glavne funkcionalnosti aplikacije, a posle toga će se izvršiti Potpuno sistemsko testiranje koje podrazumeva upravo testiranje celog sistema sa svim funkcionalnostima kao što je predviđeno planom. Predviđeno vreme trajanja ove faze je 10 radnih dana.

4.7. Faza VII – Testiranje Sigurnosti

Tokom testiranja sigurnosti proveriće se da li su podaci uneti u sistem sigurni, da li korisnici mogu da pristupe delovima sistema u skladu sa svojim pravima i da li korisnik može da izvrši nedozvoljen napad na aplikaciju. Provera sigurnosti sistema vršiće se automatski primenom alata za testiranje i analizu podataka koji primenjuje neki algoritam ili grubu silu pri pogađanju kombinacije korisničkih imena i lozinke dok ne pronađu pravu kombinaciju. Predviđeno vreme trajanja ove faze je 3 radna dana.

4.8. Faza VIII – Beta testiranje

Ovo je faza u kojoj se sprovodi testiranje nad sistemom koji ima jasno definisane funkcionalnosti. Ovo testiranje će se vršiti od strane šire grupe beta testera. Predviđeno vreme trajanja ove faze je 10 radnih dana.

5. Test podaci

Podaci koji će se koristiti u toku testiranja su:

1. Podaci koji se nalaze u bazi podataka koji su prethodno ubačeni upravo radi pripreme sistema za testiranje.
2. Podaci koje će članovi test tima unositi i toku testiranja aplikacije a koji treba da što vernije simuliraju realnu upotrebu aplikacije koja se očekuje od stvarnih korisnika u praksi.
3. Podaci koje će sistem da generiše kao rezultat matematičkih operacija koje su planirane da se vrše nad podacima unetim od strane korisnika, odnosno test tima.

Pošto se završi faza Beta testiranja, ova verzija softvera u kojoj su uklonjeni svi pronađeni problemi predstavlja kandidata za isporuku (*RC-Release Candidate*) koji će se predati klijentu.