Testiranje Projekta

si3psi

Autor: Filip Đurđević

Testira: Milanove SSU-ove , performanse

Tim: Sportaši

Dokument je namenjen za prikaz manuelnih testova,

objašnjenje automatskih i prikaz testova performansi

**1. Funkcionalno Testiranje**

Potrebno je testirati aplikaciju. Međutim, testiranje same mape nosi sa sobom određene probleme. Iz tog razloga je testiranje mape odrađeno manuelno, uz manji broj automatskih testova. Ostatak aplikacije potrebno je testirati automatski.

**1.1. SSU4 Zadavanje lokacija**

Preduslov je da je korisnik ulogovan. Zadavanje lokacija testirano je isključivo manuelno jer se radi o interakciji sa mapom. Testovi su nam pokazali da pri lošoj konekciji slika koja se prikazuje pri postavljanju krajnje lokacije može da pravi problem ukoliko njen link vodi ka nekom sajtu. Prilikom postavljanja krajnje lokacije, slika se nekad uopšte nije ni učitavala. Link je izmenjen tako da vodi ka slici na serveru. Takođe, iz nama nepoznatih razloga, dupli klik za postavljanje krajnje lokacije je prestao da radi (radio je neposredno pre predaje projekta), pa je samo zamnjen za desni klik. Ostatak funkcionalnosti se korektno izvršava.

**1.2. SSU5 Prikazivanje rute**

Prikazivanje rute testirano je i manuelno i automatski. Automatski je testirano samo dugme. Preduslov je da korisnik bude ulogovan. Sve funkcioniše kao i što je očekivano.

**1.3. SSU9 Filtriranje lokacija**

Filtriranje lokacija testirano je i automatski i ručno. Preduslov je da korisnik bude ulogovan. Prilikom testiranja otkriveno je par sitnijih i lako rešivih problema. Prvi je bio skidanje filtera zone. Prilikom ovog čina, dugme je funkcionisalo, ali nije filter nije imao efekta. To je rešeno ubicavanjem funkcije showAll() kada se skida filter. Drugi problem je bilo prikazivanje prethodno odabranog filtera konkretne zone nakon postupka skidanja i vraćanja filtera svih zona. Ovaj problem rešen je jednostavnim $(‘#checkboxN’).checked=false.

**1.4. SSU10 Pretraga mesta**

Pretraga mesta testirana je I automatski I ručno. Preduslov je da je korisnik ulogovan. Nisu pronađene nikakve greške.

**1.5. Ostali problemi**

Pronađeno je još par problema, kao što su bag kod paginacije i prikazivanja polja za izmenu lokacije, što je rešeno izbacivanjem paginacije (function over form). Takođe, moderatorska stranica nije prikazivala lokacije, pa su urađene izmene na serverskoj i klijentskoj strani. Postoje još neki sitni problemi koje za sada nije potrebno rešiti.

**2. Testiranje performansi aplikacije**

Potrebno je testirati I performanse aplikacije. Razlozi I dodatna pojašnjenja mogu se pronaći u samoj dokumentaciji projekta. U ovom dokumentu objašnjen je proces testirana performansi, zaključci pri tom procesu, kao I određeni elementi koji nam pomažu pri vizuelizaciji procesa I odnosa podataka. Uz neke stavke priložena su urađena rešenja ili moguća dodatna rešenja.

**2.1. Testiranje funkcionalnog ograničenja broja lokacija**

Broj lokacija bio je ograničen na približno 165 lokacija. Testiranje je rađeno ručno. Ograničenje je predstavljala veličina zahteva koji se šalju API-u. Rešenje je jednostavan algoritam koji šalje više zahteva I na osnovu tih zahteva sam traži najbližu lokaciju. Samim ovim procesom unosimo dodatno kašnjenje, ali to nama ne predstavlja veliki problem, kao što se može videti u Tabeli 1. Ovim smo odstranili ograničenja u pogledu API-a.

**2.2. Testiranje trajanja obrade zahteva**

Rešenje iz prethodne stavke nam omogućava bolje testiranje trajanja obrade zahteva. Umesto da se ograničimo na 165 lokacija, možemo da testiramo performanse za mnogo veći broj lokacija. Testiranje je rađeno ručno. Na osnovu zaključka da je maksimalni broj lokacija u jednom zahtevu 165, dobijamo maksimalni inkrement. Broj zahteva koji se šalju API-u biće *ceil(location\_number/increment).* U tabeli ispod ovog teksta dat je prikaz trajanja celokupnog procesa obrade zahteva (u milisekundama). Može se primetiti da nema velikog odstupanja pri porastu broja lokacija (ukoliko ima preko 600 lokacija), pa možemo da ‘‘pogađamo’’ I neke druge vrednosti. Na primer, recimo za 50 000 lokacija bi po ovoj računici proces trajao 1 minut, za 200 000 4m… U kolonama Test n prikazano je trajanje zahteva pri korišćenju OSRM API-a (API koji se trenutno koristi), dok je u koloni Prosek OSRM API prikazan prosek tih testova. Kolona Predviđeno Directions API označava približno vreme koje bi bilo potrebno za jedan korisnički zahtev da smo nastavili da koristimo isti API.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Broj lokacija | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Prosek OSRM API | Predviđeno Directions API |
| 1 | 194.351 | 138.090 | 253.323 | 201.660 | 196.126 | 196.708‬ | 200 |
| 601 | 865.39 | 1097.33 | 993.624 | 879.857 | 960.296 | 959.310 | 601 000 (~10min) |
| 901 | 1248.99 | 1452.17 | 1371.94 | 1435.44 | 1244.07 | 1350.522 | 901 000 (~15min) |
| 2059 | 2739.269 | 2632.16 | 2790.13 | 2687.78 | 2838.86 | 2737.640 | 2 059 000 (~34min) |

**Tabela 1.** Performanse pretrage najbliže lokacije