

PREDMETNI ZADATAK - INTERAKCIJA ČOVEK RAČUNAR

Opis funkcionalnosti zadatka
2024/2025. školska godina

1. VOĐENJE EVIDENCIJE O MAPI DOGAĐAJA

Napraviti jednostavnu aplikaciju za vođenje evidencije o geografskoj poziciji svetskih događaja.

Tri osnovna entiteta kojim aplikacija upravlja su: **događaj**, **tip događaja** i **etiketa (tag)**. Potrebno je realizovati distribuciju preko mape sveta na koju se prevlače i spuštaju simboli različitih događaja. Mapa je fiksna slika koja se ne skroluje i ne zumira, i studenti treba da je odaberu. Svi podaci se čuvaju u fajlu i učitavaju prilikom pokretanja aplikacije. Format fajla ostaje na izbor studentu.

Svaki **događaj** je opisan preko:

jedinstvene ljudski-čitljive oznake koju unosi korisnik,
naziva,
opisa,
tipa,
posećenosti događaja,
ikonice,
informacije da li je humanitarnog karaktera,
prosečne cene troškova za održavanje,
države i grada kao mesto održavanja događaja,
istorije datuma održavanja (lista datuma kada je događaj prethodno održavan),
datuma održavanja za tekuću godinu.

- Ikonica je sličica koja se učitava i koja se koristi da se događaj označi na mapi. Korisnik može da ne odabere ikonicu i, u slučaju kada je ne odabere, onda se podrazumevano uzima ikonica tipa kom događaj pripada.
- Posećenost događaja je jedna od sledećih vrednosti: do 1000, 1000-5000, 5000-10000 i preko 10000.
- Prosečna cena troškova je izražena u dolarima.
- Država i grad kao mesto održavanja događaja su samo podatak koji se čuva i ne utiču na položaj događaja na mapi. Korisnik sam prevlači i spušta događaj na mapu gde god on želi.
- Događaji takođe mogu biti i "tagovani" sa nijednom, jednom, ili više etiketa. **Etikete (tagove)** kreira korisnik i one su opisane svojom jedinstvenom ljudski-čitljivom oznakom koju unosi korisnik, bojom i opisom. Etiketa predstavlja "*sticky note*" za događaj i služi da čuva dodatne informacije o događaju.
- **Tip događaja** je opisan preko svoje jedinstvene ljudski-čitljive oznake koju unosi korisnik, naziva, ikonice i opisa. Ikonica je sličica koja se učitava od strane korisnika koji kreira tip i koristi se da se taj tip događaja označi na mapi.

Primer:

Tip događaja - muzički festival, filmski festival itd.

Događaj - Exit, Kustendorf itd.

Etiketa - neka dodatna informacija o događaju, napomena itd.

2. OSNOVNI ZADACI APLIKACIJE

Aplikacija treba da obezbedi:

1. Ažuriranje osnovnih podataka o događajima, tipovima i etiketama.
2. Tabelarni pregled događaja uz filtriranje i pretragu.
3. Prikaz mape i direktnu manipulaciju simbolima na mapi na pregledan način. Filtiranje simbola na mapi po nazivu događaja.

1. Ažuriranje osnovnih podataka o događajima, tipovima i etiketama

Podrazumeva opciju dodavanja, brisanja, izmene i pregleda informacija o navedenim entitetima. Pri dodavanju i izmeni događaja potrebno je omogućiti određivanje tipa događaja kao i njihovo "tagovanje" prethodno dodatim etiketama.

2. Tabelarni pregled događaja uz filtriranje i pretragu

Potrebno je da aplikacija ima tabelarni prikaz svih događaja i u okviru te tabele da se prikazuju rezultati filtriranja i pretrage.

Pretraga podrazumeva unos vrednosti jednog ili više atributa događaja prema kojima će se podaci pretraživati. Na primer, ako se unesu vrednosti za naziv i opis, potrebno je pronaći elemente koji zadovoljavaju **i jednu i drugu vrednost** istovremeno za date atribute. Ako se odabere neki tip događaja, jedna od vrednosti za posećenost i opis, potrebno je pronaći one događaje koji pripadaju tom tipu, imaju odabranu vrednost za posećenost i sadrže uneti opis. Pretraga ne mora da bude *real-time*, već je dovoljno da se pretraga, nakon unosa kriterijuma, inicira na klik dugmeta.

Filtriranje podrazumeva unos vrednosti u jedno polje, a kao rezultat se prikazuju svi događaji koji za **bilo koji** atribut imaju tu vrednost. Filtriranje je *real-time* (pri unosu svakog karaktera).

NAPOMENA: Potrebno je omogućiti da se filtriranje i pretraga ponište.

3. Prikaz mape i direktna manipulacija simbolima (ikonama) na mapi na pregledan način. Filtiranje simbola (ikonica) na mapi po nazivu događaja.

Omogućiti korisniku:

- Vizuelni pregled mape, učitane pozicije događaja iz fajla i njihove ikonice postavljene na tim pozicijama na mapi.
- Pored mape, potrebno je imati prikaz svih događaja sa kog je moguće prevući konkretan događaj na mapu. Takođe, potrebno je omogućiti i suprotno - da se događaj ukloni sa mape i vrati na prikaz svih događaja. Događaji se mogu slobodno pomerati po mapi, tj. prevlačiti sa jedne pozicije na drugu.
NAPOMENA: Događaji se ne smeju preklapati!
- Filtriranje događaja na mapi na osnovu unetog naziva događaja.
NAPOMENA: Filtriranje u tabeli i filtriranje na mapi su dve odvojene funkcionalnosti!
- Prikaz osnovnih atributa bitnih za jasnu i jedinstvenu identifikaciju događaja da bi korisnik imao predstavu sa kojim događajem manipuliše na mapi.

NAPOMENA: Zadatak **obavezno** realizovati implementacijom direktne manipulacije i *Drag&Drop* tehnike.

VAŽNE NAPOMENE:

- Na kontrolne tačke i odbranu je potrebno doneti projekat sa već kreiranim testnim podacima koji su domenski smisleni (prilagođeni temi). Na primer, unosi poput "abc" i "blabla" nisu adekvatni za naziv, opis, lokaciju događaja itd. Takođe, slike moraju biti odabrane u skladu sa temom projekta. Potrebno je da postoji barem po pet već kreiranih objekata za svaki od entiteta. Pored toga, potrebno je pripremiti još par smislenih slika i podataka (u nekom tekstualnom fajlu) za demonstraciju kreiranja, izmena i ostalih funkcionalnosti na odbrani.
 - Putanje moraju biti relativne, a format datuma i vremena nezavisan od podešavanja formata na računaru na kome se projekat pokreće. Projekti koji ne zadovoljavaju ove zahteve neće moći da se pokrenu ili neće biti potpuno funkcionalni na drugim računarima izuzev onog na kome je projekat pravljen. Ove greške mogu otežati demonstraciju i pregledanje implementiranih funkcionalnosti, a njihovo ispravljanje na odbrani zahteva dodatno vreme.
 - **Projekti koji ne zadovoljavaju pomenute zahteve neće biti pregledani.**
 - Prilikom pisanja koda, poštovati konvencije za pisanje C# i XAML koda, pisati što "čistiji" kod (engl. *clean code*). Posebnu pažnju treba obratiti na pravilno imenovanje kontrola, promenljivih, funkcija, klase itd. Jezik na kome se piše kod mora biti konzistentan kroz čitavu aplikaciju (poželjno engleski).
 - Neophodno je poznavanje i razumevanje sopstvenog koda u potpunosti i do najsitnijih detalja. Dakle, obavezno je znati šta i kako svaka linija koda u rešenju radi, tj. čemu služi. **Za neadekvatna objašnjenja, nepoznavanje i nerazumevanje sopstvenog koda, biće oduzimani bodovi.**
 - **Plagiranje i zloupotreba tuđih rešenja se strogo sankcionиšu.**
-